



HAL
open science

Approche déclarative de la modélisation de surfaces

Raphaël La Greca

► **To cite this version:**

Raphaël La Greca. Approche déclarative de la modélisation de surfaces. Interface homme-machine [cs.HC]. Université de la Méditerranée - Aix-Marseille II, 2005. Français. NNT: . tel-00011149

HAL Id: tel-00011149

<https://theses.hal.science/tel-00011149>

Submitted on 5 Dec 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE LA MÉDITERRANÉE
AIX-MARSEILLE II



FACULTÉ DES SCIENCES DE LUMINY
163, Avenue de Luminy 13288 MARSEILLE cedex 9

THÈSE DE DOCTORAT
en Informatique

Approche déclarative de la modélisation de surfaces

présentée par :

Raphaël LA GRECA

en vue d'obtenir le grade de docteur de l'Université de la Méditerranée



soutenue le 31 Octobre 2005 devant le jury composé de :

Marc DANIEL	Professeur, Directeur de thèse
Véronique GAILDRAT	Maître de conférence (HDR), Rapporteur
Jean-Claude LÉON	Professeur, Rapporteur
Dominique MICHELUCCI	Professeur, Examineur
Marc NEVEU	Professeur, Président du jury
Jean SEQUEIRA	Professeur, Examineur
Philippe VÉRON	Maître de conférence (HDR), Examineur

N° attribué par la bibliothèque

*À ma famille,
À Aurore, ma tendre épouse,
«Aux enfants qui viendront,
qui nous ressembleront . . .»*

REMERCIEMENTS

Je commencerai par remercier l'ensemble des membres du jury pour avoir tous accepté de se déplacer ce lundi 31 octobre 2005 sur Marseille en cette veille de jour férié. Je leur suis extrêmement reconnaissant pour l'attention toute particulière qu'ils ont portée à mon travail,

- Marc NEVEU, pour m'avoir fait l'honneur de présider mon jury de thèse et pour tous les axes de recherche qu'il a su m'ouvrir à travers ses questions sur les perspectives d'évolution de mon étude aux surfaces de subdivision ;
- Véronique GAILDRAT, pour avoir accepté la tâche de rapporteur, pour la relecture attentive de mon rapport de thèse ainsi que pour l'ensemble des remarques constructives qu'elle a su apporter à mon travail ;
- Jean-Claude LÉON, pour avoir accepté la tâche de rapporteur ainsi que pour m'avoir donné la possibilité d'utiliser le logiciel DEFSURF développé par son équipe de recherche au laboratoire 3S¹ de Grenoble ;
- Dominique MICHELUCCI, pour son importante implication dans l'Action Spécifique CNRS *modélisation géométrique sous contraintes* et le projet *MathSTIC* dans lesquels se sont intégrés mes travaux, ainsi que pour ses conseils avisés accompagnés de ceux de Pascal SCHRECK pour l'organisation des journées associées qui se sont déroulées à Marseille ;
- Jean SEQUEIRA, pour m'avoir accueilli dans son équipe de recherche (LXAO²), pour toute la confiance qu'il a su m'accorder durant ces quelques années de thèse et pour les moyens qui m'ont été donnés pour réaliser ce travail ;
- Philippe VÉRON, pour son suivi et sa collaboration précieuse dans nos travaux sur la description de pièces mécaniques dans le cadre du pôle *modélisation géométrique* avec l'équipe IMS³ ;
- Marc DANIEL, que je ne pourrai jamais assez remercier tant son investissement a été important. Qu'il trouve ici toute ma reconnaissance pour avoir dirigé mes recherches, pour ses remarques, ses critiques et ses conseils qui ont toujours été constructifs et d'une grande qualité, pour ses encouragements de chaque instant, pour tout le temps qu'il a pu me consacrer en jours, nuits, vacances et week-ends à lire et relire mes travaux malgré un emploi du temps très chargé, pour sa patience, sa loyauté, son honnêteté et sa rigueur ainsi que pour toutes les autres raisons qui m'ont permis de mieux connaître le monde de la recherche et de l'apprécier.

J'adresse tous mes remerciements à Alexandra BAC pour son aide précieuse dans les démonstrations mathématiques et sa grande disponibilité.

¹Sols, Solides et Structures

²Modélisation, Conception et Reconstruction Assistées par Ordinateur

³Ingénierie, Mécanique, Systèmes

Je tiens à remercier Vincent CHEUTET et Jean-Philippe PERNOT pour leur aide dans la manipulation du logiciel DEFSUF qu'ils développent.

J'adresse tout particulièrement mes remerciements à l'équipe INCA⁴, en pensant à Laurent HENOCQUE, Mathieu ESTRATA et Nicolas PROVIC pour notre collaboration dans le cadre du pôle *modélisation géométrique* autour de la sémantique et de la description.

Je souhaite remercier mes collègues arlésiens, et plus particulièrement les membres locaux de l'équipe LXAO : Gilles GESQUIÈRE, Romain RAFFIN, Eric REMY et Sébastien THON pour leur aide, leurs bons conseils et nos discussions toujours plus enrichissantes.

J'adresse un grand merci à tous les membres de l'équipe LXAO qui m'a accueilli et que j'ai eu la chance de côtoyer. C'est avec un grand plaisir que j'ai travaillé à leur côté et que j'ai partagé leur enthousiasme et beaucoup de moments mémorables!

Je n'oublie pas tous mes amis et amies qui ont toujours été présents à mes côtés, je leur en serai toujours reconnaissant.

Toute ma gratitude va à ma famille qui malgré mon absence parfois très longue, n'a pas cessé de me montrer son affection et son soutien. J'embrasse mes parents et mon adorable sœur pour leurs encouragements et leur amour de chaque instant. Je souhaite aussi remercier tout particulièrement mon grand-père Ange CAPITANI pour m'avoir donné depuis tout petit, cette passion pour les sciences et la recherche. Je n'oublie pas ma belle-famille : M. et Mme. LORDT et Grégory (mon coach sportif), pour toute l'affection qu'ils me portent.

Enfin, je ne pourrai jamais assez remercier ma tendre épouse, Aurore, qui a fait preuve d'une patience exemplaire, qui a toujours été là pour m'aider notamment par ses relectures, pour toutes ses petites attentions comme ses délicieux petits plats, et qui a su faire tous les lourds sacrifices qu'implique l'accompagnement de la vie d'un doctorant.

À tous, encore merci ...

Raphaël

⁴Inférence, Contraintes et Applications

TABLE DES MATIÈRES

Table des matières	i
Table des figures	vii
Introduction	1
1. Approche déclarative appliquée à la modélisation géométrique	5
1.1 Introduction	5
1.2 La modélisation géométrique	5
1.3 Les surfaces en modélisation géométrique	6
1.3.1 Les modèles implicites	7
1.3.2 Les modèles polyédriques	9
1.3.3 Les modèles de subdivision	10
1.3.4 Les modèles paramétriques	13
1.3.4.1 Paramétrisation d'une surface	14
1.3.4.2 Notations	16
1.3.4.3 Propriétés	17
1.3.4.4 Notions de courbure	18
1.4 Concept déclaratif	19
1.4.1 Historique	19
1.4.2 Les fondements de la modélisation déclarative	21
1.4.2.1 Description de la scène	22
1.4.2.2 Recherche des solutions : génération	22
1.4.2.3 Prise de connaissance des solutions	23
1.4.2.4 Modeleur traditionnel vs Modeleur déclaratif	23
1.4.3 Problématique et besoins de la modélisation de surfaces dans une optique déclarative	25
1.4.4 Approches possibles de la modélisation déclarative de surfaces	26
1.5 Détails de l'approche proposée	27
1.5.1 La méthode	27
1.5.2 L'idée	28
1.5.3 L'approche conceptuelle	29
1.6 Conclusion	31

2. Description de surfaces	33
2.1 Introduction	33
2.2 Les moyens de description	34
2.3 Importance du contexte et orientation métier	37
2.4 Les besoins en vocabulaire pour la description de surfaces	38
2.4.1 La localisation	39
2.4.1.1 Éléments référents de la scène	41
2.4.1.2 Localisation relative (ou déictique)	42
2.4.1.3 Localisation absolue	43
2.4.2 La géométrie	46
2.4.3 La topologie	46
2.4.4 La morphologie	46
2.4.5 Les termes de comparaisons	47
2.4.6 Les éléments de jonctions	47
2.4.7 Les contraintes	47
2.4.8 La quantification	48
2.5 Étude de la description dans un contexte «Mécanique»	49
2.5.1 Les objectifs	49
2.5.2 Protocole d'expérimentation	50
2.5.3 Première phase	51
2.5.4 Deuxième phase	53
2.5.5 Troisième phase	55
2.6 Conclusion	56
3. Modèles sémantiques	57
3.1 Introduction	57
3.2 De la description vers la sémantique	58
3.2.1 Le moteur de traduction sémantique	58
3.2.2 Structure du modèle sémantique et de ses généralisations	60
3.2.3 La base de connaissances	67
3.2.3.1 Le graphe de dépendances fonctionnelles et déformations disponibles	67
3.2.3.2 Les localisations	69
3.2.3.3 Les formes prédéfinies	70
3.2.3.4 Les quantificateurs	72
3.2.4 La complétion sémantique	74
3.2.4.1 Complétion symbolique	75
3.2.4.2 Complétion numérique	76
3.2.4.3 Application sur un exemple simple	78
3.3 Le moteur de planification	84

3.3.1	Notion de classe de solutions : le modèle patron	84
3.3.2	Planification à partir d'un modèle sémantique généralisé	88
3.3.2.1	Analyse post-génération	89
3.3.2.1.1	Évolution de \mathcal{A}_s en \mathcal{G}_{dsh}	90
3.3.2.1.2	Construction du graphe de dépendances inter-zones \mathcal{G}_{diz}	91
3.3.2.2	Génération des modèles patrons	97
3.4	Conclusion	103
4.	Construction des modèles géométriques	105
4.1	Introduction	105
4.2	Méthodes de déformation existantes	105
4.2.1	Approches géométriques	106
4.2.1.1	Manipulation directe	106
4.2.1.2	Manipulation indirecte par volume de déformation	108
4.2.2	Approches physiques et mécaniques	111
4.2.3	Approches multirésolutions	115
4.3	Méthodes de déformation proposées	117
4.3.1	Utilisation de DEFSURF	117
4.3.2	Évaluation des quantificateurs	120
4.3.3	Raffinement des régions à déformer	121
4.3.4	Déformation par zone : contraintes «lâches»	127
4.3.4.1	Vecteur normal à un point de contrôle	127
4.3.4.2	Vecteur normal moyen à plusieurs points de contrôle	128
4.3.4.3	Plan moyen à plusieurs points de contrôle	128
4.3.4.4	Aplanir, Tendre ou Lisser une zone	129
4.3.4.5	Aplatir une zone	130
4.3.4.6	Extruder ou Estamper / Bomber/Creuser une zone	132
4.3.4.7	Gonfler/Dégonfler une zone	135
4.3.4.8	Trouer une zone	137
4.3.4.9	Bosseler une zone	138
4.3.4.10	Courber, plier, corner ou enrouler une zone	139
4.3.5	Satisfaction d'une contrainte ponctuelle avec contrôle d'influence	147
4.3.5.1	Application aux courbes NURBS	147
4.3.5.1.1	Détermination du paramètre u	149
4.3.5.1.2	Influence de la fonction $f(i)$	149
4.3.5.2	Application aux surfaces NURBS	153
4.3.6	Généralisation à N contraintes ponctuelles avec contrôle d'influence	156
4.3.6.1	Application aux courbes	156
4.3.6.1.1	Notations	156

4.3.6.1.2	Méthode 1 : Approche globale	158
4.3.6.1.3	Expression algébrique de la méthode globale	160
4.3.6.1.4	Méthode 2 : Approche séquentielle	163
4.3.6.1.5	Expression algébrique de la méthode séquentielle	165
4.3.6.1.6	Comparaison des 2 méthodes	169
4.3.6.2	Application aux surfaces	170
4.3.6.2.1	Notations	170
4.3.6.2.2	Méthode 1 : Approche globale	171
4.3.6.2.3	Méthode 2 : Approche séquentielle	172
4.3.6.2.4	Formulation algébrique et liens entre les méthodes	174
4.3.7	Gestion de certaines contraintes «lâches» comme cas particuliers de contraintes «dures»	175
4.3.8	Fonctionnement global du moteur de résolution/construction	177
4.4	Conclusion	178
5.	Résultats & Applications	181
5.1	Introduction	181
5.2	Les applications développées	181
5.2.1	La bibliothèque <i>R3D</i>	182
5.2.2	La bibliothèque <i>RDS</i>	182
5.2.3	La maquette de modéleur déclaratif : « <i>Declarative NURBS designer</i> »	183
5.2.4	Le sculpteur de surfaces : « <i>NURBS carver</i> »	185
5.3	Quelques résultats	188
5.3.1	Exemple 1 : Capot de Twingo®	189
5.3.1.1	Cas de la première description	191
5.3.1.2	Cas de la deuxième description	193
5.3.2	Exemple 2 : Carénage de moto	193
5.4	Conclusion	198
	Synthèse et perspectives	199
	Synthèse	199
	Perspectives	200
A.	Modélisation géométrique	203
A.1	Géométrie différentielle	203
A.1.1	Première forme fondamentale	203
A.1.2	Deuxième forme fondamentale	203
A.1.3	Courbures principales	204
A.1.4	Courbure gaussienne	204
A.1.5	Courbure moyenne	205
A.1.6	Courbure absolue	205

A.2	Géométrie discrète	205
A.2.1	Complexe simplicial	205
A.2.2	Validité de maillage	206
A.2.2.1	La relation d'Euler	206
A.2.2.2	Contrainte de défaut angulaire	206
A.2.2.3	Être en 2-variété sans trou ni tétraèdre résiduel	207
B.	La description : les premiers pas d'une approche déclarative	209
B.1	Première expérience sur la description de surfaces	209
B.2	Deuxième expérience sur la description de formes	210
B.2.1	Descriptions de formes obtenues et résultats associés	210
B.2.1.1	Pièce en «U» (phase 1)	211
B.2.1.1.1	Description spontanée	211
B.2.1.1.2	Description structurée	211
B.2.1.1.3	Résultats	212
B.2.1.1.4	Organigramme de description	212
B.2.1.2	«Cendrier» (phase 1)	213
B.2.1.2.1	Description spontanée	213
B.2.1.2.2	Description structurée	213
B.2.1.2.3	Résultat	214
B.2.1.2.4	Organigramme de description	214
B.2.1.3	L'«évier» (phase 1)	215
B.2.1.3.1	Description spontanée	215
B.2.1.3.2	Description structurée	215
B.2.1.3.3	Résultat	216
B.2.1.3.4	Organigramme de description	216
B.2.1.4	L'«île» (phase 2)	217
B.2.1.4.1	Description structurée	217
B.2.1.4.2	Résultat	217
B.2.1.5	Capot de Twingo® (phase 2)	218
B.2.1.5.1	Description structurée	218
B.2.1.5.2	Résultat	218
B.2.1.6	Rétroviseur de formule Renault TM (phase 2)	219
B.2.1.6.1	Description structurée	219
B.2.1.6.2	Résultat	219
B.2.2	Détail du vocabulaire utilisé	220
B.2.2.1	Classification par type de vocabulaire avec nature grammaticale	220
B.2.3	Réflexions sur des grammaires de descriptions possibles	222
B.2.3.1	Première grammaire de description	222

B.2.3.2	Deuxième grammaire de description	223
C.	La sémantique utilisée	225
C.1	Syntaxe des modèles sémantiques	225
C.2	Base de connaissances	226
C.3	Un modèle sémantique généralisé pour le capot de Twingo®	228
D.	Algorithmes proposés	231
D.1	Algorithme de raffinement du polyèdre de contrôle	231
D.2	Satisfaction de N contraintes ponctuelles sur une courbe NURBS selon la méthode 1 dite globale	232
D.3	Satisfaction de N contraintes ponctuelles sur une courbe NURBS selon la méthode 2 dite séquentielle	233
E.	Preuves et démonstrations	235
E.1	Preuve de convergence de la méthode globale pour l'influence naturelle des contraintes	235
E.2	Méthode séquentielle appliquée aux courbes : Preuve de la proposition 4.3.8	239
E.3	Méthode séquentielle appliquée aux courbes : Preuve de la proposition 4.3.9	240
	Bibliographie	243

TABLE DES FIGURES

1	Vision multidisciplinaire de la modélisation déclarative	2
1.1	Trois primitives ponctuelles et l'iso-surface associée	7
1.2	Squelette et iso-surface d'un embranchement et d'un chromosome	8
1.3	Complexe simplicial et les k -simplexes le composant	9
1.4	Schémas de subdivision proposés par Charles Loop en 1987	11
1.5	Les schémas de subdivision de Doo-Sabin, Calmull-Clark et Loop	12
1.6	Polyèdre de contrôle de \mathcal{S}	14
1.7	Plan paramétrique \mathcal{P}	14
1.8	Surface paramétrique \mathcal{S}	14
1.9	Restriction paramétrique \mathcal{R}_1	15
1.10	Restriction paramétrique \mathcal{R}_2	15
1.11	Surface restreinte suivant \mathcal{R}_1 puis \mathcal{R}_2	15
1.12	PolyFormes	20
1.13	VoluFormes	20
1.14	MultiFormes : La collégiale du Dorat	20
1.15	MultiFormes : La collégiale du Dorat texturée	20
1.16	Processus de conception en spirale	21
1.17	Comparaison de modeleurs : traditionnel vs déclaratif	24
1.18	Niveaux de description et techniques associées	26
1.19	Niveaux d'abstraction de la méthode MERISE	27
1.20	Processus déclaratif général proposé	28
1.21	Architecture du processus déclaratif proposé	29
2.1	Arbuste défini par un L-System	35
2.2	Bâtiment défini par un FL-System	35
2.3	Courbe fermée décrite par « $M^+0m^-M^-m^-0M^+m^+M^-m^+$ »	35
2.4	Surface à décrire	38
2.5	Les catégories de vocabulaire	39
2.6	Le vocabulaire du lexique	40
2.7	Référentiel absolu par défaut du volume d'une scène	41
2.8	Localisations relatives à un objet A dans un plan borné carré	42
2.9	Localisations absolues des lignes et points usuels d'un plan borné carré	43
2.10	Localisations absolues des zones usuelles d'un plan borné carré	44
2.11	Localisations absolues dans un volume selon les 3 axes	45
2.12	Localisations absolues affiniées dans un volume	45

2.13	Pièce en «U»	49
2.14	«Cendrier»	49
2.15	«Évier»	49
2.16	«île»	50
2.17	«Capot de Twingo [®] »	50
2.18	«Rétroviseur de formule Renault TM »	50
2.19	Schéma de validation d'une description	51
2.20	Résultats pour la forme : pièce en «U»	52
2.21	Diagramme de description de la pièce en «U»	54
2.22	Résultats pour la forme : capot de Twingo [®]	55
3.1	Moteur de traduction sémantique et ses modules d'interfaçage	58
3.2	Arbre de déclinaison des modèles sémantiques ou arbre des solutions	60
3.3	Schéma UML de la structure d'un modèle sémantique généralisé	61
3.4	Arbre structurant identique pour des localisations différentes de zone	62
3.5	Plan paramétrique \mathcal{R}_0 associé à \mathcal{S}	63
3.6	Exemple de zone à forme trouée : \mathcal{Z}_1	65
3.7	Résultat correspondant à l'exemple de modèle sémantique généralisé	66
3.8	Extrait du graphe de dépendances fonctionnelles de la base de connaissances	68
3.9	Localisation «ANGLE HG» et ses différentes configurations	69
3.10	Polygones prédéfinis d'une forme «CARREE» (a) et «RONDE» (b)	70
3.11	Courbe NURBS circulaire de degré 2 et définie par 7 pôles	71
3.12	Graphe des probabilités et pourcentages attachés aux quantificateurs	73
3.13	Du modèle sémantique vers ses généralisations : la complétion sémantique	74
3.14	Ex. 1 : Arbre structurant	78
3.15	Ex. 1 : Arbre des solutions après la complétion symbolique de \mathcal{Z}_0	81
3.16	Non commutativité des déformations	84
3.17	Étape 2 : prise en compte des chevauchements entre zones	85
3.18	Moteur de planification	88
3.19	Graphes utilisés durant le processus de planification	89
3.20	Étape 1 : prise en compte de la hiérarchie	90
3.21	Ex 1 : Plan paramétrique localisant les zones et graphe \mathcal{G}_{dsh} résultat	91
3.22	Ex 2 : Arbre structurant et plan paramétrique avec frontières des zones	93
3.23	Ex 2 : de \mathcal{A}_s vers \mathcal{G}_{dsh}	94
3.24	Ex 2 : extrait de \mathcal{G}_{df} contenu dans la base de connaissances	94
3.25	Exemple : Graphe de dépendances inter-zones	95
3.26	Exemples de configurations de type «Chaîne fermée»	96
3.27	Exemples de configurations de type «Chaîne ouverte»	96
3.28	Construction des modèles patrons, étape d'initialisation	100
3.29	Construction des modèles patrons, choix d'un nœud à traiter	100
3.30	Ex 2 : Graphe de séquences de construction	103

TABLE DES FIGURES

4.1	Interpolation : satisfaction d'une contrainte ponctuelle et d'une contrainte tangentielle	107
4.2	Approximation par minimisation du carré des distances	107
4.3	Aplatissement d'une courbe [Piegl and Tiller, 1997]	108
4.4	Modélisation par déformations de forme libre	108
4.5	Exemple de déformation de forme libre étendue	109
4.6	Déformation par surfaces d'influence	109
4.7	Déformation par axe courbe et par surface quadratique	110
4.8	Ajustement optimal d'une Surface D-NURBS	111
4.9	Modèle mécanique constitué de masses, ressorts et charnières	112
4.10	Méthode de déformation de surfaces gauches	113
4.11	DEFSURF : Utilisation de lignes cibles et ligne de limitation	114
4.12	DEFSURF : Impact de la fonctionnelle sur la déformation	114
4.13	Représentation multirésolution d'une courbe B-Spline	115
4.14	Déformation d'une courbe à Pôles en multirésolution	116
4.15	Arbre des modèles géométriques d'un modèle patron	117
4.16	Analogie d'une ligne frontière de zone avec une ligne de limitation	118
4.17	Construction possible d'une ligne cible pour la déformation «Bombée»	118
4.18	Déformation «Bombée» avec DEFSURF	119
4.19	Variation de la fonction de quantification selon la valeur du poids	121
4.20	Surface plane avec prise en compte des propriétés de la forme	123
4.21	Surface plane avec prise en compte des restrictions de forme	123
4.22	Raffinement de plusieurs zones d'une même surface	124
4.23	Association des pôles influents à chaque zone de la surface	125
4.24	Les zones de contrôle d'une surface	126
4.25	Cas d'une zone de contrôle trouée sur la surface	126
4.26	Normale à un pôle	127
4.27	Construction du <i>plan moyen</i> à un ensemble de pôles	129
4.28	Aplanir, tendre ou lisser une zone	130
4.29	Principe du traitement « <i>aplatir</i> » appliqué sur une courbe	131
4.30	Aplatissement d'une zone	131
4.31	Fonction de contrôle de la déformation <i>bomber</i>	132
4.32	Principe du traitement « <i>bomber</i> » appliqué sur une courbe	133
4.33	Bombement d'une zone	133
4.34	Principe du traitement « <i>extruder</i> » appliqué sur une courbe	135
4.35	Extrusion d'une zone	135
4.36	Principe du traitement « <i>gonfler</i> » appliqué sur une courbe	136
4.37	Gonflement d'une zone	137
4.38	Trouer une zone de la surface	137
4.39	Principe du traitement « <i>bosseler</i> » appliqué à une courbe	138

4.40	Bosseler une zone	139
4.41	Exemples de zones pouvant être courbées, accompagnées de leur axe paramétrique de courbure	140
4.42	Exemples de zones ne pouvant pas être courbées	140
4.43	Projection des images paramétriques des pôles sur $A_p(1)$	141
4.44	Définition des intervalles de travail pour $C_e=3$	141
4.45	Conditions initiales d'une déformation «courber»	143
4.46	Ex. de déformation «courber» : traitement de l'intervalle 1	143
4.47	Ex. de déformation «courber» : traitement de l'intervalle 2	144
4.48	Ex. de déformation «courber» : traitement de l'intervalle 3	144
4.49	Courber une zone	145
4.50	Enroulement d'une zone	145
4.51	Plier une zone	146
4.52	Exemple de contrainte de passage sur une courbe	147
4.53	Satisfaction d'une même contrainte spatiale selon trois contraintes de localité	149
4.54	Exemple de filtre gaussien	150
4.55	Influence de plusieurs filtres gaussiens sur une même contrainte	151
4.56	Filtrage gaussien d'une image par convolution matricielle	152
4.57	Contrainte ponctuelle de passage sur une surface	153
4.58	Contrainte ponctuelle de passage ayant des influences différentes	154
4.59	Surface : Impact de la taille d'un filtre gaussien sur la satisfaction d'une même contrainte	155
4.60	Déroutement pas à pas de la méthode globale sur une courbe	159
4.61	Interaction des contraintes de localisation sur une courbe	159
4.62	Méthode globale : Convergence dans le cas d'une surcontrainte paramétrique	160
4.63	Déroutement pas à pas de la méthode séquentielle sur une courbe	164
4.64	Déroutement pas à pas de la méthode globale sur une surface	172
4.65	Déroutement pas à pas de la méthode séquentielle sur une surface	174
4.66	Résultat de la formulation algébrique sur une surface	174
4.67	Exemples de déformation : «La zone est bombée et passe par le point A» .	176
4.68	Notion de contraintes <i>floues</i> pour la résolution de contraintes de proximité .	177
4.69	Principe du moteur de Résolution/Construction	177
5.1	Applications et bibliothèques utilisées	182
5.2	<i>Declarative NURBS designer</i> : fenêtre de <i>commandes</i>	184
5.3	<i>Declarative NURBS designer</i> : fenêtre de <i>rendu</i>	184
5.4	<i>Declarative NURBS designer</i> : console des opérations effectuées	185
5.5	<i>NURBS carver</i>	186
5.6	Résultat après satisfaction des contraintes avec <i>NURBS carver</i>	187
5.7	Manipulation de courbes gauches avec <i>NURBS carver</i>	188
5.8	Forme de la surface que l'on souhaite obtenir : un Capot de Twingo®	189

TABLE DES FIGURES

5.9 Arbres structurants correspondants aux deux descriptions	190
5.10 Définition des zones dans le plan paramétrique	191
5.11 Instance possible du modèle patron 1	191
5.12 Instance possible du modèle patron 2	192
5.13 Instance possible du modèle patron 3	192
5.14 Instance possible du modèle patron 4	192
5.15 Instance possible correspondant à la deuxième description	193
5.16 Modèle de la surface du carénage que l'on souhaite obtenir	194
5.17 Définitions des zones du carénage dans le plan paramétrique	194
5.18 Hiérarchie des zones	195
5.19 Instance possible du modèle patron 1	195
5.20 Instance possible du modèle patron 2	196
5.21 Instance possible du modèle patron 3	196
5.22 Instance possible du modèle patron 4	196
5.23 Instances des modèles patrons avec la nouvelle déformation « <i>creuser</i> » . . .	197
5.24 Modèle patron 4 vu de l'intérieur	198
A.1 Complexe simplicial pur et non-pur	206
A.2 Voisinage et défaut angulaire d'un sommet S_i	207
B.1 Résultats pour la forme : pièce en «U»	212

INTRODUCTION

La Conception Assistée par Ordinateur, ou CAO, est aujourd'hui un domaine essentiel de l'informatique pour l'industrie. Elle permet en effet de construire et visualiser des objets et des pièces à fabriquer ou bien de tester et de simuler des comportements qui pourraient être longs, dangereux et coûteux s'ils devaient être véritablement mis en œuvre. La CAO est donc, aujourd'hui, pour les entreprises, un outil de travail permettant de lier efficacité, rapidité et économie.

Dans cette thèse, nous nous intéressons à la conception de surfaces et plus particulièrement aux modèles de surfaces paramétriques de type NURBS⁵. Le choix de ce type de surface a été motivé par le fait qu'elles sont très largement utilisées dans le monde de la conception de pièces mécaniques, domaine majeur pour les applications de la CAO actuellement. Ces surfaces sont construites et manipulées par l'intermédiaire d'un polyèdre de contrôle qui n'est autre qu'une grille définie par un ensemble de pôles. Chacun de ces points de contrôle exerce sur la surface continue qu'il maîtrise, une attraction permettant de la modeler.

Dans les approches classiquement utilisées dans ce domaine, les concepteurs réalisent des pièces mécaniques ou des objets par l'association d'un ensemble de carreaux de ces surfaces. Dans la majorité des cas, la manipulation d'un de ces carreaux nécessite un nombre de points de contrôle important à gérer, ce qui rend le travail des concepteurs très long, difficile et mal adapté à l'activité créatrice à laquelle ils doivent s'atteler. Pour pallier ces difficultés et aider les concepteurs dans leur tâche, plusieurs logiciels mettent à disposition des outils permettant de manipuler des ensembles de pôles. Mais leur utilisation reste très attachée au modèle mathématique sous-jacent des surfaces, guidant ainsi le cheminement constructif du concepteur et le freinant dans sa démarche créatrice.

L'approche que nous proposons repose sur le concept *déclaratif* de la modélisation. Cette notion qui a émergé à la fin des années 80 à Nantes, consiste à réaliser et construire des objets, des formes, des images ou même des scènes par la simple donnée d'un ensemble de caractéristiques et de propriétés à vérifier. Par ce principe, le système déclaratif détermine l'ensemble des éléments solutions répondant aux exigences de l'utilisateur. Celui-ci n'a, alors, plus qu'à choisir le ou les résultats qui se rapprochent le plus de l'image mentale de ce qu'il souhaite obtenir. Cette approche permet ainsi au concepteur de se libérer des

⁵Non Uniform Rational B-Spline

contraintes attachées aux modèles manipulés et de se consacrer pleinement à son activité créative. En plus de ces côtés attractifs pour l'utilisateur, l'approche déclarative enrichit les modèles de la connaissance apportée lors de la description. Ces informations peuvent être utilisées dans les traitements futurs des éléments conçus afin d'accélérer les processus ou bien de travailler dans des environnements contraints à respecter les conditions initiales de construction.

La mise en œuvre d'un tel système nécessite la manipulation de nombreux domaines d'application attachés à l'informatique. La modélisation déclarative est ainsi considérée comme un carrefour entre plusieurs disciplines comme le montre la figure 1.

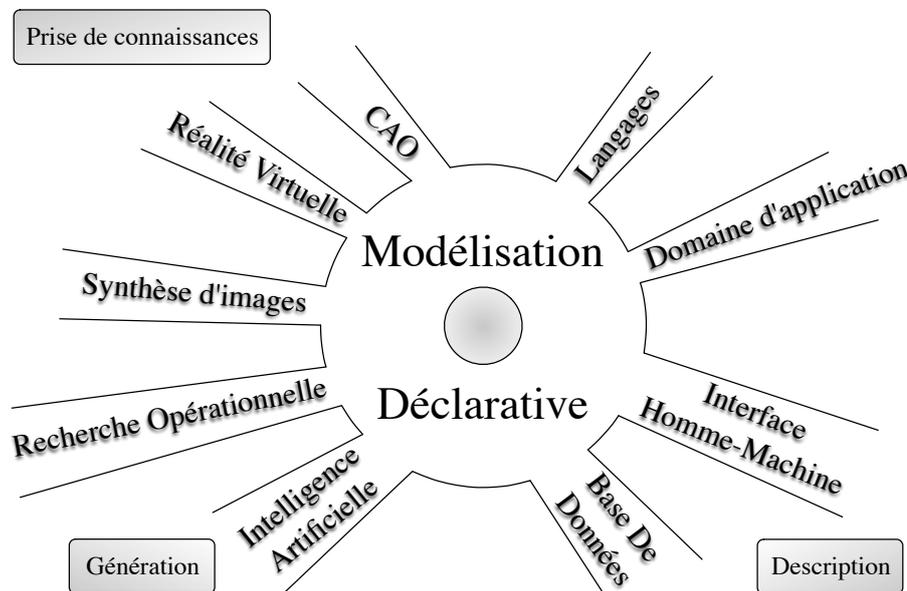


Figure 1. Vision multidisciplinaire de la modélisation déclarative [Champciaux, 1998, Gaildrat, 2003]

L'approche déclarative de la modélisation de surfaces est donc un sujet très large. C'est pourquoi, l'étude que nous proposons dans ce document se limite à la conception d'un simple carreau de surface NURBS. Les quatre principaux objectifs de cette thèse sont les suivants :

1. étudier la faisabilité de l'approche appliquée aux surfaces ;
2. concevoir une architecture complète du processus à réaliser ;
3. identifier les besoins et les perspectives d'évolution du système ;
4. réaliser une maquette permettant d'obtenir des résultats simples de surfaces utilisant le système pensé.

Afin d'y parvenir, le présent document se divise en cinq chapitres. Le premier est un chapitre introductif, où la problématique et chacun des termes clés du sujet sont précisément définis. Il présente donc le concept de *modélisation* en CAO et détaille l'approche dite *déclarative* par l'énoncé des travaux du domaine. Ce même chapitre décrit les modèles de surface utilisés en modélisation géométrique et s'attarde plus particulièrement sur les surfaces NURBS. Enfin, il précise l'architecture conceptuelle globale du système qui sera en même temps le plan général de ce document.

Le second chapitre traite de l'aspect descriptif des surfaces. Il montre qu'il est possible de décrire des surfaces et des formes en utilisant un langage proche du langage naturel. Il décrit le vocabulaire qui peut être utilisé à cet effet ainsi que l'ensemble des travaux que nous avons réalisés autour du langage de description.

Le chapitre trois est réservé à l'étude sémantique de notre système. Il introduit donc la structure des modèles sémantiques employés ainsi que la base de connaissances déclarative utilisée. Il détaille aussi tous les traitements qui sont rattachés à cette sémantique et qui permettent d'aboutir à des classes de solutions.

Le chapitre quatre constitue une partie importante du document. Nous y abordons les notions de déformations de modèles surfaciques à pôles, de contraintes *lâches* et de contraintes *dures*, avant d'y présenter la technique que nous avons élaborée pour obtenir des modèles géométriques répondant aux résultats des chapitres précédents.

Le cinquième et dernier chapitre propose quelques résultats géométriques obtenus et présente rapidement les applications qui ont été menées à être développées durant cette thèse.

Enfin nous concluons en donnant toutes les perspectives de recherche et d'évolution pour compléter l'étude réalisée.

APPROCHE DÉCLARATIVE APPLIQUÉE À LA MODÉLISATION GÉOMÉTRIQUE

1.1 Introduction

Ce premier chapitre introduit l'étude menée dans le cadre de cette thèse. Il s'articule autour des trois mots clés du sujet que sont la *modélisation*, les *surfaces* et l'*approche déclarative*. Ainsi, la première partie de ce chapitre donne une définition de la *modélisation* dans son contexte géométrique et fait émerger la notion de *surface*. Plusieurs modèles géométriques permettant la représentation de surfaces y sont présentés en précisant leurs points forts, leurs points faibles ainsi que leurs domaines d'application privilégiés. Nous insistons ensuite plus particulièrement sur les surfaces paramétriques puisque ce sont les modèles géométriques qui sont actuellement les plus utilisés en CAO, domaine vers lequel s'oriente l'ensemble des travaux réalisés au cours de cette thèse. La deuxième partie de ce chapitre met en avant le concept «*déclaratif*». L'introduction de ce terme central de l'étude nous amène aux besoins de la CAO dans cette direction. Enfin dans la quatrième et dernière partie, nous présentons la structure générale de l'approche que nous proposons en survolant toutes les notions qui seront ensuite détaillées point par point dans la suite de ce document.

1.2 La modélisation géométrique

Dans les approches systémiques [Guinier, 1992], la *modélisation* est vue comme «une représentation d'un système réel» qui a pour vocation d'être plus intelligible, favorisant ainsi la compréhension de systèmes complexes. Un *modèle*, issu de la modélisation, est alors une représentation d'un système qui vise à en faciliter la compréhension. Il met en évidence certains aspects ou certaines parties et en ignore d'autres : «Un modèle est d'autant plus abstrait qu'il ignore plus les détails de la réalité» [Planche, 1988]. Dans son approche géométrique, le principe de la modélisation est sensiblement le même puisque son objectif est la représentation virtuelle (ou numérique) d'un objet réel. Les modèles géométriques représentent la surface frontière (ou le volume) d'un objet ou d'une forme. Ils doivent être le plus proche possible de la réalité qu'ils cherchent à reproduire. Ainsi un objet peut avoir

une représentation plus ou moins complexe ou plus ou moins simplifiée selon l'application pour laquelle il est destiné. De tels modèles peuvent être utilisés dans les domaines de la simulation, de la réalité virtuelle ou de la visualisation scientifique que l'on retrouve à travers :

- La CFAO¹ : objets ou pièces mécaniques ;
- L'architecture ;
- Les jeux vidéos ;
- Le cinéma : effets spéciaux, films d'animation ;
- La médecine : modélisation d'organes et chirurgie assistée ;
- La simulation.

Dans le cadre de notre étude, nous nous penchons plus particulièrement sur la modélisation géométrique appliquée au domaine de la CAO. Celle-ci se définit comme l'ensemble des outils et techniques informatiques utilisés en vue de la réalisation de produits manufacturables. Elle est utilisée dans l'ingénierie mécanique comme dans l'architecture et peut être vue comme l'équivalent de la table à dessin d'un ingénieur ou d'un architecte associée à une très grande puissance de calculs permettant, entre autre, de gérer un grand nombre de contraintes dès la phase de conception. La CAO fait son apparition dans l'industrie dans les années 1970 et devient très vite un outil indispensable. Elle a beaucoup évolué et permet aujourd'hui d'assister la conception d'une pièce en intégrant de la connaissance aux modèles manipulés. La notion de «*features*» ou formes caractéristiques fait aujourd'hui partie intégrante des systèmes de CAO et permet à travers la sémantique sous-jacente d'accroître ses performances et ses domaines d'application. Ainsi, la CAO se révèle de plus en plus fiable, de plus en plus précise et de plus en plus rapide, ce qui répond parfaitement à la mutation des systèmes de production à l'heure d'une mondialisation grandissante : «*Faire moins cher, plus vite, des systèmes de bonne qualité, sans polluer*» [Tollenaere et al., 1998]. Dans le contexte de cette thèse, à travers la modélisation géométrique et la CAO, c'est la notion de *surface* qui nous intéresse principalement. C'est pourquoi nous la détaillons dans la partie suivante.

1.3 Les surfaces en modélisation géométrique

La modélisation géométrique permet la représentation virtuelle d'objets et de formes par la représentation de leurs surfaces frontières. Ainsi, afin de mieux percevoir ce que l'on entend par surface en modélisation géométrique, cette partie présente les grandes catégories de modèles [Léon, 1991, Foley and al., 1997, Farin, 2002] permettant leur représentation : les modèles **implicites**, les modèles **polyédriques**, les modèles de **subdivision** et les modèles **paramétriques**. Chacun de ces modèles est ici présenté en insistant sur ses avantages, sur ses inconvénients et sur le domaine d'application privilégié de son exploitation.

¹Conception et Fabrication Assistée par Ordinateur

1.3.1 Les modèles implicites

Un modèle implicite S , ou surface implicite [Bloomenthal and al., 1997], se définit dans sa forme générale comme une *iso-surface* d'un champ de potentiels $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ par :

$$S = \{ \mathbf{P} \in \mathbb{R}^3 \mid F(\mathbf{P}) = iso \}, \text{ avec } iso \in \mathbb{R}$$

Ce champ de potentiels F , aussi appelé fonction implicite, contrôle la forme de la surface. Son choix constitue donc la principale difficulté pour une manipulation intuitive par l'utilisateur.

La fonction implicite F peut être définie de façon analytique, ainsi la surface implicite représentant la sphère unité s'écrit :

$$F(\mathbf{P}) = \mathbf{P}_x^2 + \mathbf{P}_y^2 + \mathbf{P}_z^2 - 1 = 0$$

et celle permettant d'obtenir des super-quadriques [Barr, 1981] s'écrit :

$$F(\mathbf{P}) = \left(\left(\frac{\mathbf{P}_x}{a_1} \right)^{\frac{2}{\varepsilon_1}} + \left(\frac{\mathbf{P}_y}{a_2} \right)^{\frac{2}{\varepsilon_1}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{\mathbf{P}_z}{a_3} \right)^{\frac{2}{\varepsilon_1}}$$

Où ε_1 et ε_2 permettent de contrôler l'aspect anguleux de la super-quadrique.

L'ensemble des formes engendrées reste cependant difficile à manipuler même s'il est possible de les déformer [Sclaroff and Pentland, 1991].

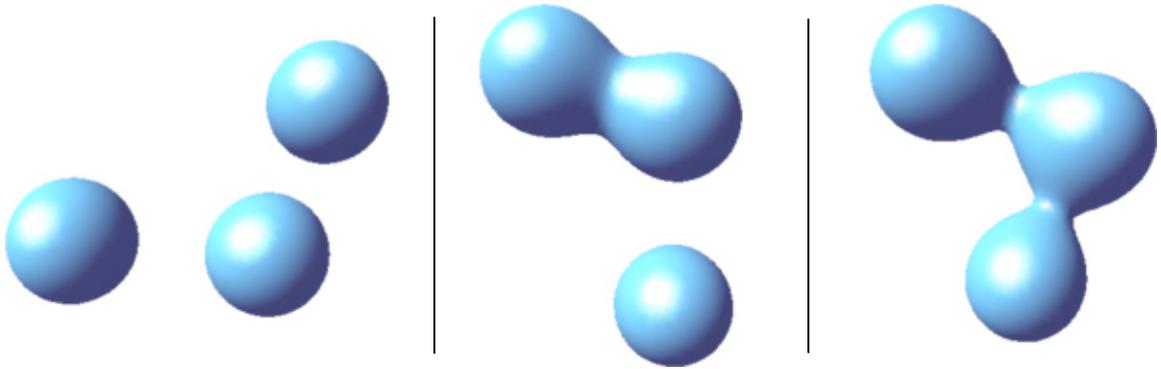


Figure 1.1. Trois primitives ponctuelles et l'iso-surface associée

L'approche implicite dite par squelette permet alors d'obtenir des formes plus complexes. Le squelette n'étant rien d'autre qu'un ensemble de primitives géométriques sur lesquelles sont définis des potentiels. Ces primitives peuvent être des points (Fig. 1.1), des segments, des triangles ou même des éléments plus complexes comme des courbes de subdivision [Hornus and al., 2003]. La fonction implicite F devient la somme des fonctions F_i de chaque primitive. La figure 1.2 montre deux exemples de la modélisation implicite par squelette. Le lecteur pourra se référer à la thèse de Jean-Luc Mari [Mari, 2002] pour rentrer plus en détail dans l'aspect théorique de cette technique.

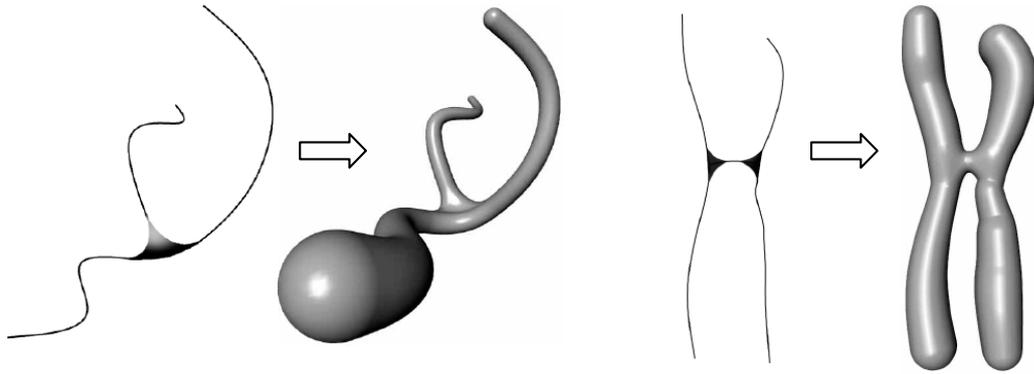


Figure 1.2. Squelette et iso-surface d'un embranchement et d'un chromosome [Ferley and al., 1997]

Les **avantages** majeurs apportés par l'utilisation de surfaces implicites sont les suivants :

- La rapidité de mise en œuvre ;
- Une continuité au moins C^1 qui confère aux surfaces une très bonne qualité visuelle ;
- Le squelette, constitué de primitives, contrôle l'aspect morphologique de la surface ;
- La facilité pour différencier l'intérieur de l'extérieur d'une surface fermée, par convention :
 - si $F(P) > iso$ alors P est à l'intérieur de S ;
 - si $F(P) = iso$ alors P est sur S ;
 - si $F(P) < iso$ alors P est à l'extérieur de S .
- La normale N en tout point $P \in S$ se calcule simplement par :

$$N(\mathbf{P}) = \frac{\nabla F(\mathbf{P})}{\|\nabla F(\mathbf{P})\|} \text{ où } \nabla F(\mathbf{P}) = \left(\frac{\partial F}{\partial x}(\mathbf{P}), \frac{\partial F}{\partial y}(\mathbf{P}), \frac{\partial F}{\partial z}(\mathbf{P}) \right)$$

Les principaux **inconvénients** attachés aux surfaces implicites sont les suivants :

- La représentation de formes complexes implique un squelette formé d'un grand nombre de primitives ;
- Les fonctions implicites sont difficiles à déterminer pour obtenir l'effet exact désiré ;
- On ne peut que s'approcher de la représentation des arêtes vives [Pasko and al., 1995, Wyvill and van Overveld, 1996] indispensables en CAO ;
- Les surfaces obtenues sont des formes fermées ;
- Il n'existe pas nécessairement de paramétrisation d'une surface implicite, il est donc très difficile de se déplacer à sa surface ;
- La visualisation de surfaces implicites passe par des méthodes coûteuses comme les *marching cubes* [Lorensen and Cline, 1987] ou le lancé de rayons [Sherstyuk, 1999].

Les surfaces implicites ne sont pas particulièrement bien adaptées pour une utilisation dans le monde de la CAO. Ces modèles sont et restent les plus adaptés au domaine de l'animation [Cani, 1999] où il suffit d'appliquer un mouvement au squelette de la forme pour que celui-ci soit, par définition, transmis à la surface en conservant son lissage.

1.3.2 Les modèles polyédriques

Un modèle polyédrique [Boissonnat et Yvinec, 1995] , aussi appelé maillage, est une représentation discrète de la surface d'un objet ou d'une forme [Kobbelt and al., 2000]. Cette architecture se décompose en éléments finis appelés *sommets*, *arêtes* et *faces*. Une face est polygonale et est construite à partir de plusieurs arêtes. Une arête est quant à elle l'association de deux sommets.

Bien que les faces puissent être des polygones quelconques, elles sont parfois choisies quadrangulaires mais restent le plus fréquemment triangulaires. Ce choix s'explique d'une part parce que les facettes triangulaires ont le grand avantage d'être planes mais aussi parce que le maillage peut alors se définir comme un *complexe simplicial* dont les propriétés mathématiques sont connues. Ces notions sont approfondies en annexe A.2.

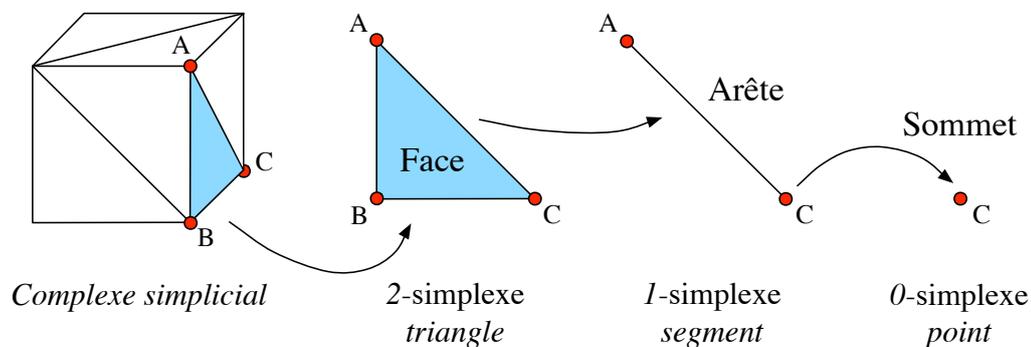


Figure 1.3. Complexe simplicial et les k -simplexes le composant

Les modèles polyédriques ont de nombreux **avantages**, voici les plus importants :

- Simplicité de mise en œuvre ;
- De par sa structure discrète, les modèles polyédriques sont très adaptés à la visualisation par ordinateur. Plusieurs bibliothèques graphiques² proposent d'ailleurs l'affichage de primitives simpliciales en temps réel ;
- Les arêtes vives sont simples à réaliser ;
- L'architecture des données représentant la structure maillée peut conserver l'information de voisinage de chaque sommet. Il est ainsi possible de se repérer sur la surface de l'objet modélisé [Clouchoux and al., 2004] ;
- La version à faces triangulaires des modèles polyédriques a de nombreuses propriétés connues ce qui l'implique dans de nombreux algorithmes du domaine de la géométrie discrète comme les problèmes de simplification de maillage [Schroeder and al., 1992, Garland and Heckbert, 1997] ou bien de reconstruction et de triangulation de nuages de points [Hoppe, 1994, Amenta and al., 1998, Randrianarivony and Brunnett, 2002, Attali and Boissonnat, 2003].

²OpenGL, DirectX ...

En contrepartie, les modèles polyédriques ont aussi des **inconvenients** :

- Les modèles polyédriques sont construits à partir d’un grand nombre de primitives qui ne fait qu’augmenter avec la précision ce qui les rend vite volumineux pour de gros objets ou des formes complexes ;
- Du fait de son grand nombre de primitives, les manipulations et les déformations des modèles polyédriques sont parfois difficiles et nécessitent des outils spécifiques [Borrel and Rappoport, 1994, Raffin, 2000, Botsch and Kobbelt, 2004] ;
- La représentation d’objets courbes très lisses est assez difficile sans un nombre important de faces ;
- La surface de l’objet représenté n’est que C_0 continue ;
- Le conditionnement et la qualité de la polygonalisation du maillage ont de très fortes répercussions sur le rendu esthétique de l’objet modélisé ;
- Le calcul des normales et des courbures aux sommets du polyèdre ne sont que des estimations. Ces calculs de courbures discrètes [Maltret et Daniel, 2001, La Greca, 2001, Maltret and Daniel, 2002b, Maltret and Daniel, 2002a] sont particulièrement sensibles à la qualité de la polygonalisation du maillage.

Bien que les modèles polyédriques soient parfois utilisés en CAO pour la modélisation dans leur version triangulaire ou quadrangulaire, ils restent les modèles incontestablement destinés à la visualisation. Les modèles implicites et les modèles paramétriques sont d’ailleurs, la plupart du temps, visualisés sous la forme de modèles polyédriques après leur discrétisation.

1.3.3 Les modèles de subdivision

Par leur structure, les modèles de subdivision s’appuient sur les modèles polyédriques. Ils ont été introduits dès 1978 [Catmull and Clark, 1978, Doo and Sabin, 1978] inspirés des travaux de Chaikin [Chaikin, 1974] travaillant sur le raffinement de courbes.

Le principe général de la génération de modèles de subdivision consiste à appliquer des opérations de raffinement du maillage par l’ajout ou la modification de sommets. Chaque opération est conditionnée par des règles de subdivision aussi appelées **schémas de subdivision**.

Les schémas de subdivision ont tous des propriétés. Certaines sont spécifiques à un schéma, d’autres sont communes à plusieurs d’entre eux. Nous présentons ici en quelques mots les propriétés les plus importantes :

La propriété de *localité* est commune à tous les schémas. Elle indique que la position d’un nouveau sommet à insérer dans un maillage se calcule à partir d’un nombre fini de sommets identifiés par le schéma de subdivision.

Chaque schéma est caractérisé par ses *règles de subdivision* qui indiquent comment doit être inséré un nouveau sommet dans le maillage ou comment doit être modifié un sommet existant en fonction de son voisinage. Les sommets impliqués dans le schéma sont associés à

des coefficients qui permettent par leur somme pondérée de positionner le sommet concerné. Deux grands types de schémas sont à dissocier, les schémas *d'interpolation* et les schémas *d'approximation*. Les premiers ont la propriété de conserver la position des sommets du maillage initial aussi appelé maillage de contrôle. Ils n'ont donc pas de règles de subdivision visant à modifier des sommets existants. Les seconds, ou schémas d'approximation, comportent au moins une règle de modification de sommets, les sommets du maillage de contrôle ne sont donc pas conservés et le maillage résultat ne passe pas forcément par les sommets initiaux.

La *valence* d'un sommet est le nombre de sommets auxquels il est connecté. Cette notion rejoint celle de voisinage abordée dans la section 1.3.2. Dans un schéma de subdivision, cette notion est très importante puisque c'est elle qui détermine la règle de subdivision à utiliser. Les figures 1.4.A et 1.4.B présentent d'une part une configuration de valence 6 décrivant une règle de modification de sommet et d'autre part trois configurations de règles d'insertion de sommet sur les arêtes. Les pondérations indiquées ont été proposées par Charles Loop en 1987 et permettent une convergence de la surface de subdivision vers une surface B-Spline de degré 2. Ainsi, chaque schéma de subdivision possède une valence privilégiée pour laquelle il a un comportement stable. Les sommets dont la valence est privilégiée par le schéma sont dit **réguliers**, les autres sont appelés sommets **extraordinaires**.

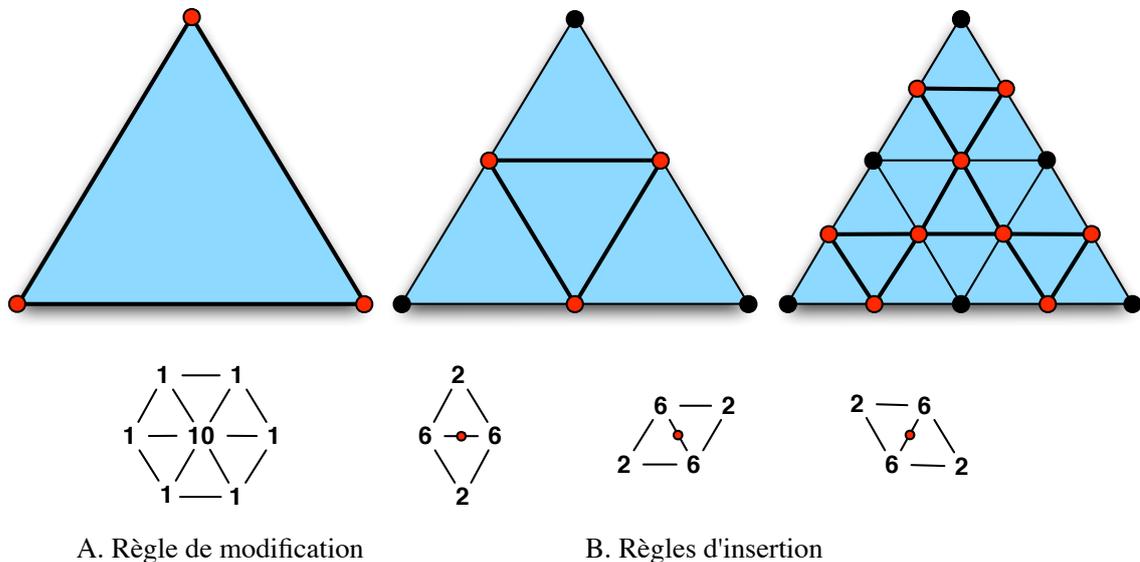


Figure 1.4. Schémas de subdivision proposés par Charles Loop pour des surfaces triangulaires et pondérations pour tendre vers des B-Splines de degré 2 [Loop, 1987]

Certains schémas ont fait l'objet d'extensions [Barthe and Kobbelt, 2004] par l'ajout de règles supplémentaires adaptées à la configuration de sommets extraordinaires. Bien que ces règles *extraordinaires* permettent d'apporter de meilleurs résultats, elles sont très souvent insuffisantes. Le meilleur moyen d'obtenir de bons résultats reste alors de travailler sur des maillages d'un type adapté au schéma de subdivision utilisé comportant le moins

possible de sommets extraordinaires.

De nombreux schémas de subdivisions ont été mis au point comme celui de Catmull-Clark [Catmull and Clark, 1978], de Loop [Loop, 1987], de Doo-Sabin [Doo and Sabin, 1978], celui dit de Butterfly [Dyn and al., 1996] en référence à sa forme ou encore celui de Kobbelt [Kobbelt, 2000] aussi connu sous le nom de subdivision par $\sqrt{3}$. Le tableau suivant [Astart, 2004] fait état des 3 schémas de subdivision les plus courants en donnant le type de maillage sur lesquels ils s'appliquent et leur type de surface limite. La figure 1.5 permet de visualiser les résultats qui peuvent être obtenus :

Schéma	Type de maillage	Type de surface limite
Doo-Sabin	Polygonal	B-Spline quadratique
Catmull-Clark	Polygonal	B-Spline cubique
Loop	Triangulaire	B-Spline triangulaire quadratique

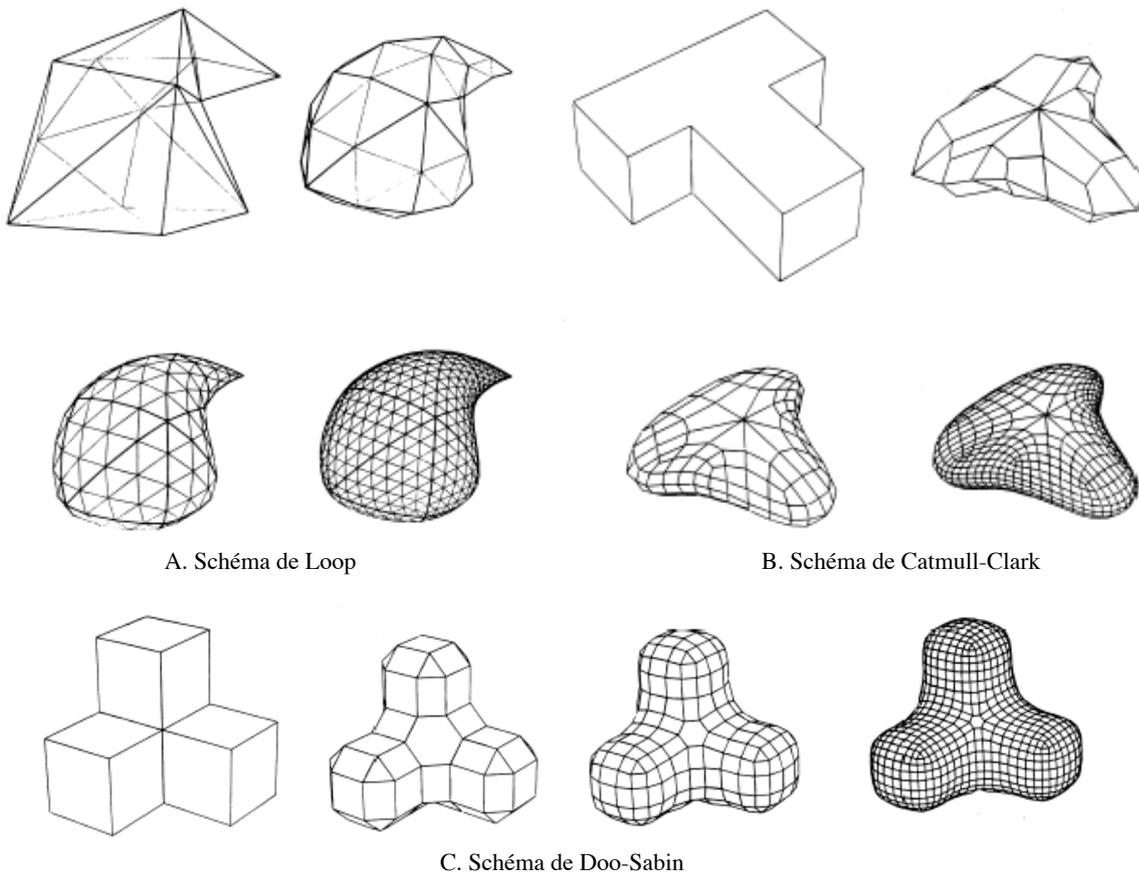


Figure 1.5. Les schémas de subdivision de Doo-Sabin, Calmull-Clark et Loop

Les modèles de subdivision reposent sur les modèles polygonaux, ils ont donc les mêmes avantages et les mêmes inconvénients. Il est tout de même possible d'ajouter certains points

qui leur sont propres. Commençons par les **avantages** majeurs :

- Le traitement d’objets de topologie quelconque est réalisable ;
- La multi-résolution des objets modélisés est possible selon le nombre d’itérations choisi [Lounsbery and al., 1997] ;
- Un nombre faible d’itérations permet d’obtenir des surfaces visuellement très lisses.

Mais les surfaces de subdivision ont aussi leurs **inconvénients** :

- Le nombre de faces augmente de façon exponentielle entre chaque itération ;
- La stabilité du schéma de subdivision utilisé dépend du type de maillage et de la présence de sommets extraordinaires ;
- À l’exception du schéma de Catmull-Clark [Stam, 1998], on ne connaît généralement que le type de surface limite engendré par un schéma de subdivision donné. La surface limite exacte n’est pas connue de façon analytique. Il n’est donc pas possible de faire des calculs exacts se rapportant à la surface limite comme ceux des normales par exemple.

Les modèles de subdivision commencent à véritablement émerger dans le domaine de la modélisation géométrique avec les possibilités techniques croissantes. Ils sont principalement utilisés par les studios de cinéma (Cf. PIXAR) pour la réalisation de films d’animations, d’effets spéciaux ou de jeux vidéos. La possibilité de gérer rapidement et facilement la multirésolution des modèles en fonction de la distance à la caméra est en effet un atout majeur en terme de temps de calcul et de rendu.

Les surfaces de subdivision sont cependant encore en marge de la CAO bien que beaucoup de modeleurs s’y intéressent de très près pour leur capacité intrinsèque de multi-résolution.

1.3.4 Les modèles paramétriques

L’approche déclarative de la modélisation de surfaces pourrait s’appliquer aux modèles implicites par une construction déclarative des squelettes et des champs de potentiels. Elle pourrait aussi très certainement s’appliquer aux maillages de surfaces de subdivision, cependant comme nous l’expliquons dans le chapitre suivant, la thèse s’est focalisée sur le domaine de la mécanique en CAO. Le choix du type de surfaces à utiliser dans notre étude s’est alors porté sur les surfaces paramétriques [Daniel, 1989, Demengel, 1998], et plus particulièrement sur des surfaces NURBS³ [Piegl and Tiller, 1997, Rogers, 2000, Farin, 2002], modèles géométriques actuellement les plus utilisés dans le monde de la CAO [Tollenaere et al., 1998]. Ces modèles se construisent à partir d’un ensemble de points de contrôle qui rendent plus intuitives leurs manipulations. Cette partie est entièrement dédiée à l’étude des surfaces NURBS et aux principales notions qui leur sont rattachées.

³Non Uniform Rational B-Splines

1.3.4.1 Paramétrisation d'une surface

Soit une surface \mathcal{S} (Fig. 1.8), variété de dimension 2, définie sur un plan paramétrique \mathcal{P} (Fig. 1.7) borné tel que $\mathcal{S}(u, v) \in \mathbb{R}^3$ avec $(u, v) \in [a, b] \times [c, d]$, $(a, b, c, d) \in \mathbb{R}^4$, $a < b$ et $c < d$ pouvant être ramené à $(u, v) \in [0, 1]^2$.

La surface \mathcal{S} se caractérise par un ensemble de points de contrôle ou pôles qui jouent le rôle d'attracteurs et déterminent ainsi sa forme générale. Ces points de contrôle s'organisent selon une grille et définissent le polyèdre de contrôle de \mathcal{S} (Fig. 1.6).

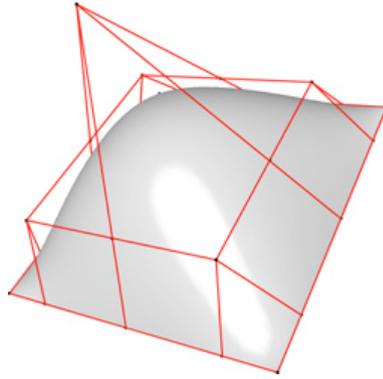


Figure 1.6. Polyèdre de contrôle de \mathcal{S}

Si u et v sont fixés en u_0 et v_0 , on obtient deux courbes dites isoparamétriques sur \mathcal{S} . Soit M le point de \mathcal{S} défini en (u_0, v_0) tel que $M = \mathcal{S}(u_0, v_0)$, M se trouve alors à l'intersection de ces deux courbes isoparamétriques. Les tangentes en ce point, \vec{M}_u et \vec{M}_v eq. (1.1), forment par définition le plan tangent à \mathcal{S} en un point régulier M ($\vec{M}_u \times \vec{M}_v \neq 0$). Ce plan est noté $(\vec{MP}, \vec{M}_u, \vec{M}_v) = 0$ avec P un point de ce plan.

$$\vec{M}_u = \frac{\partial \vec{OM}}{\partial u} \qquad \vec{M}_v = \frac{\partial \vec{OM}}{\partial v} \qquad (1.1)$$

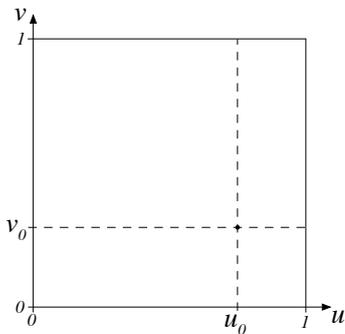


Figure 1.7. Plan paramétrique \mathcal{P}

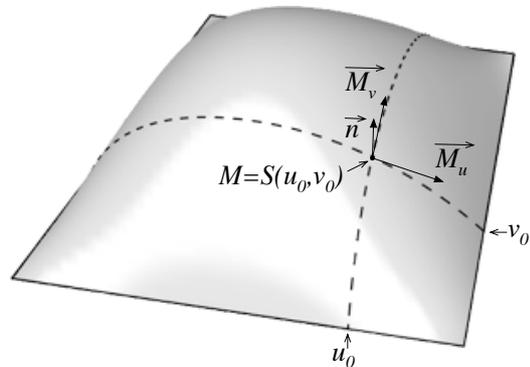


Figure 1.8. Surface paramétrique \mathcal{S}

Cependant compte tenu des bornes du domaine paramétrique, ces surfaces sont seulement constituées de quatre courbes frontières. La construction de surfaces aux frontières de formes complexes n'est donc pas possible alors que c'est la plupart du temps le cas des surfaces utilisées dans l'industrie automobile par exemple. Pour contourner cet obstacle, une solution consiste à construire une surface plus large et de définir une courbe décrivant ses frontières.

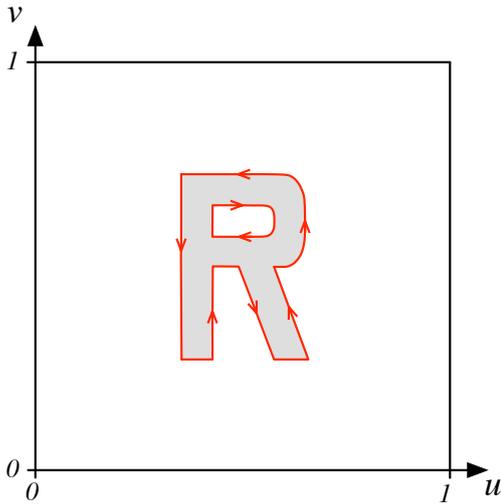


Figure 1.9. Restriction paramétrique \mathcal{R}_1

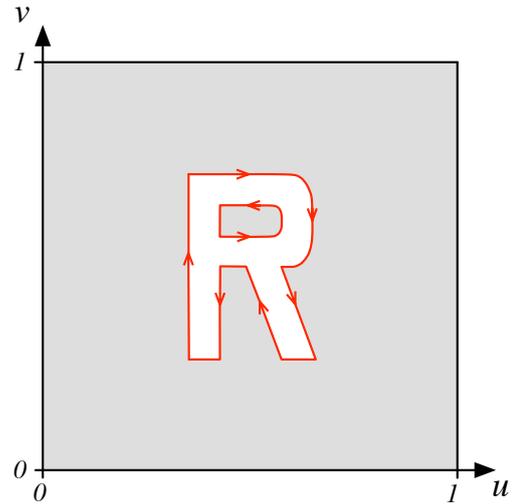


Figure 1.10. Restriction paramétrique \mathcal{R}_2

Cette courbe, définie dans le domaine paramétrique de la surface est nommée courbe de *restriction* et la surface résultat est nommée surface *restreinte*. Les courbes de *restrictions* du domaine paramétrique représentent alors les frontières de la surface *restreinte*.

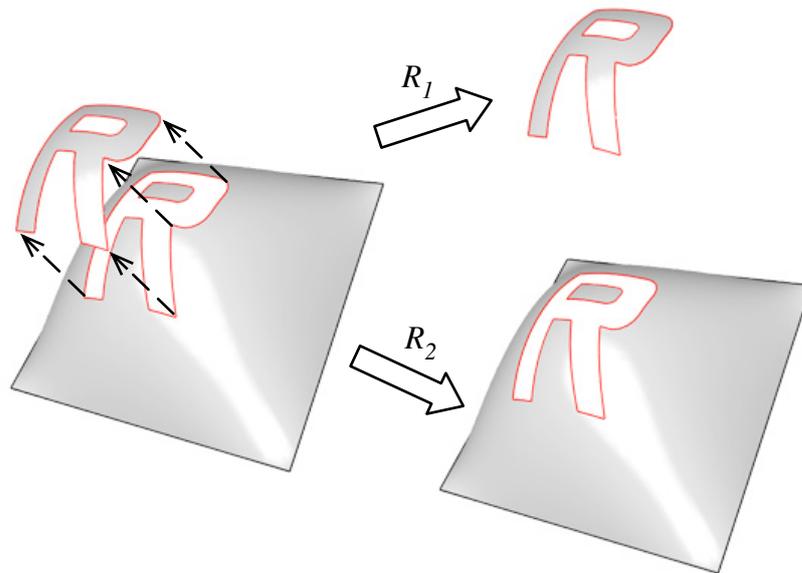


Figure 1.11. Surface restreinte suivant \mathcal{R}_1 puis \mathcal{R}_2

Deux types de frontières sont à dissocier, les frontières de la surface qui sont classiquement définies dans le sens de parcours trigonométrique (Fig. 1.9) et les frontières intérieures à la surface qui vont représenter des trous (Fig. 1.10) et qui sont eux définis dans le sens trigonométrique inverse. Il est aussi possible de combiner les deux techniques sur un même plan paramétrique afin d'obtenir des restrictions plus complexes pour une même surface. La figure 1.11 donne un exemple de surfaces restreintes à partir de ces deux restrictions. Il est possible d'assembler plusieurs de ces carreaux de surface restreints selon leurs nouvelles frontières afin de construire un objet manufacturable. Le degré global de continuité dépend des modèles surfaciques utilisés (Bézier, B-Spline, NURBS, ...) ainsi que des méthodes de connexions [Barsky and DeRose, 1990] entre les différentes surfaces (C^0 , G^1 ou C^1 continu, G^2 ou C^2 continu).

1.3.4.2 Notations

Durant tout ce document nous utilisons la notation de [Piegl and Tiller, 1997] telle qu'une surface NURBS de degré p dans la direction u et de degré q dans la direction v s'écrit sous la forme :

$$\mathcal{S}(u, v) = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} N_{i,p}(u) N_{j,q}(v) w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad \text{avec } (u, v) \in [0, 1]^2$$

où

- n est le nombre de points de contrôle dans la direction u ;
- m est le nombre de points de contrôle dans la direction v ;
- $\{\mathbf{P}_{i,j}\}$ sont les points de contrôle formant le polyèdre de contrôle ;
- $\{w_{i,j}\}$ sont les poids de chacun de ces points. Nous admettons dans notre étude que $\forall (i, j) \in \mathbb{R}^2, w_{i,j} > 0$;
- $\{N_{i,p}(u)\}$ et $\{N_{j,q}(v)\}$ sont les fonctions B-Splines (eq. 1.2) définies sur les vecteurs de noeuds :

$$U = \underbrace{\{0, \dots, 0\}}_{p+1} \underbrace{\{u_{p+1}, \dots, u_{r-p-1}\}}_{n-p-1} \underbrace{\{1, \dots, 1\}}_{p+1} \quad V = \underbrace{\{0, \dots, 0\}}_{q+1} \underbrace{\{v_{q+1}, \dots, v_{s-q-1}\}}_{m-q-1} \underbrace{\{1, \dots, 1\}}_{q+1}$$

$\underbrace{\hspace{15em}}_{p+n+1} \qquad \underbrace{\hspace{15em}}_{m+q+1}$

avec $r = n + p + 1$ et $s = m + q + 1$

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & \text{si } u_i \leq u \leq u_{i+1} \\ 0 & \text{sinon} \end{cases} \\ N_{i,p}(u) = \frac{u - U_i}{U_{i+p} - U_i} N_{i,p-1}(u) + \frac{U_{i+p+1} - u}{U_{i+p+1} - U_{i+1}} N_{i+1,p-1}(u) \end{cases} \quad (1.2)$$

Posons :

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^{n-1} \sum_{l=0}^{m-1} N_{k,p}(u) N_{l,q}(v) w_{k,l}} \quad (1.3)$$

La surface peut s'écrire :

$$\mathcal{S}(u, v) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} R_{i,j}(u, v) \mathbf{P}_{i,j} \quad (1.4)$$

1.3.4.3 Propriétés

De par sa définition algébrique, l'expression $R_{i,j}$ vérifie l'ensemble des propriétés suivantes :

1. La **non-négativité** : $\forall (i, j) \in [0, n-1] \times [0, m-1], \forall (u, v) \in [0, 1]^2, R_{i,j}(u, v) \geq 0$;
2. La **partition de l'unité** : $\forall (u, v) \in [0, 1]^2, \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} R_{i,j}(u, v) = 1$;
3. Le **support local** : $R_{i,j}(u, v) = 0$ si $(u, v) \notin [u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$;
4. Dans tout rectangle donné de la forme $[u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$, au plus $(p+1)(q+1)$ fonctions de base sont non nulles, en particulier les $R_{i,j}(u, v)$ pour $i_0 - p \leq i \leq i_0$ et $j_0 - q \leq j \leq j_0$;
5. Les **extrema** : si $p > 0$ et $q > 0$, alors $R_{i,j}(u, v)$ atteint exactement une valeur maximale ;
6. $R_{0,0}(0, 0) = R_{n-1,0}(1, 0) = R_{0,m-1}(0, 1) = R_{n-1,m-1}(1, 1) = 1$;
7. La **dérivabilité** : à l'intérieur des rectangles formés par les lignes de noeuds de u et v , toutes les dérivées partielles de $R_{i,j}(u, v)$ existent. Pour un noeud de u (respectivement v), où k est la multiplicité de ce noeud, la dérivée reste continue après $p - k$ (respectivement $q - k$) dérivations dans la direction u (respectivement v) ;
8. $\forall (i, j) \in [0, n-1] \times [0, m-1], \forall a \neq 0$, si $w_{i,j} = a$ alors $\forall (u, v) \in [0, 1]^2, R_{i,j}(u, v) = N_{i,p}(u)N_{j,q}(v)$.

Ces propriétés permettent de déduire les propriétés géométriques suivantes qui s'appliquent aux surfaces NURBS :

9. L'interpolation des points aux coins :
 $\mathcal{S}(0, 0) = P_{0,0}, \mathcal{S}(1, 0) = P_{n-1,0}, \mathcal{S}(0, 1) = P_{0,m-1}$ et $\mathcal{S}(1, 1) = P_{n-1,m-1}$;
10. l'**invariance affine** : une transformation affine s'applique sur la surface en l'appliquant à ses points de contrôle ;
11. Propriété de l'**enveloppe convexe** : $\forall (i, j) \in [0, n-1] \times [0, m-1]$ avec $w_{i,j} > 0$, si $(u, v) \in [u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$ alors $\mathcal{S}(u, v)$ est dans l'enveloppe convexe des points de contrôle $P_{i,j}$ avec $i_0 - p \leq i \leq i_0$ et $j_0 - q \leq j \leq j_0$;
12. La **modification locale** : si $P_{i,j}$ bouge ou $w_{i,j}$ change, cela n'affecte la forme de la surface que dans la zone définie par $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$;
13. La **dérivabilité** : la dérivée de $\mathcal{S}(u, v)$ reste continue après $p - k$ (respectivement $q - k$) dérivations par rapport à u (respectivement v) pour un noeud de multiplicité k ;

14. Les surfaces de Bézier, les surfaces de Bézier rationnelles et les surfaces B-Spline non rationnelles sont des cas particuliers de surfaces NURBS :

Les surfaces de *Bézier* [Bezier, 1970, Bezier, 1974] ont été introduites par Pierre Bézier dans les années 1970. Dans chaque direction, le nombre de pôles constituant le polyèdre de contrôle d'une surface de Bézier est égal au degré de cette direction plus 1. Elles sont aujourd'hui encore beaucoup utilisées pour la représentation de surfaces à l'aspect visuel très lisse. Les surfaces de Bézier deviennent rationnelles si l'on utilise des poids attachés à chacun de ses pôles. Une surface construite à partir de plusieurs carreaux élémentaires de Béziens est une surface *B-Splines* non rationnelle. Elle est alors équivalente à une surface NURBS définie sur des vecteurs de nœuds uniformes dont le poids de chaque point de contrôle est égal à l'unité ($\forall (i, j) \in [0, n - 1] \times [0, m - 1], w_{i,j} = 1$). Une surface B-Spline non-rationnelle peut donc s'écrire :

$$\mathcal{S}(u, v) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} N_{i,p}(u) N_{j,q}(v) P_{i,j} \quad (1.5)$$

1.3.4.4 Notions de courbure

En CAO, et plus particulièrement lors de la conception de surfaces, la notion de courbure est primordiale, que ce soit pour des raisons d'esthétique avec la régularité des lignes de reflets ou bien pour des raisons physiques avec des calculs d'aérodynamique par exemple. Trois types de courbures existent et permettent d'appréhender la forme locale de S en un point M . Ce sont les deux *courbures principales*, $k1$ et $k2$ déduites de la deuxième forme fondamentale (Cf. A.1.3), qui permettent de les calculer :

- la courbure *gaussienne* qui se calcule en faisant le produit de $k1$ et $k2$;
- la courbure *moyenne* qui se calcule en faisant la moyenne de $k1$ et $k2$;
- la courbure *absolue* qui se calcule en faisant la somme des carrés de $k1$ et $k2$.

Un bref rappel de ces notions est disponible dans l'annexe de *géométrie différentielle* en A.1. Pour de plus amples détails, le lecteur pourra se reporter à plusieurs ouvrages traitant du sujet : [Valiron, 1945, Do Carmo, 1976, Braemer et Kerbrat, 1976, Postnikov, 1990, Demengel, 1998, Doss-Bachelet et al., 2000].

Tout au long de ce document nous nous appuierons sur les surfaces NURBS que nous avons choisies comme modèle géométrique de surfaces pour traiter notre problématique. La partie suivante de ce chapitre en introduit d'ailleurs la thématique principale qui est l'approche *déclarative*.

1.4 Concept déclaratif

1.4.1 Historique

Le concept déclaratif a commencé à se développer à Nantes au sein de l'équipe de M. LUCAS à la fin des années 1980 après plusieurs constats :

- certaines constructions sont très longues à réaliser avec les méthodes existantes ;
- il est parfois difficile de trouver toutes les constructions qui répondent à plusieurs critères identiques ;
- certaines constructions simples à décrire sont très difficiles à obtenir avec les outils existants.

Le terme de «Modélisation Déclarative» apparaît alors pour la première fois à la fin de l'année 1989 à Strasbourg dans [Lucas et al., 1990]. On y trouve une première définition de M. Lucas qui s'adapte à la conception de formes : pour lui, *«l'objectif de la modélisation déclarative de formes est de permettre d'engendrer des formes (ou des ensembles de formes) par la simple donnée d'un ensemble de propriétés ou de caractéristiques. L'ordinateur est chargé d'explorer l'univers des formes potentielles afin de sélectionner celles correspondant à la définition donnée. Le concepteur n'a plus qu'à choisir, à l'aide d'outils appropriés, la ou les formes qui lui conviennent»*. Il peut alors se concentrer sur l'aspect créatif de sa réalisation sans avoir à rentrer dans les considérations mathématiques des modèles utilisés [Lucas, 1993].

En 1990, dans [Le Goff, 1990], D. Le Goff propose une autre définition à la modélisation déclarative en disant qu'elle *«repose sur l'idée que nous pouvons appréhender le "monde" autrement que par sa description géométrique : nous pouvons le percevoir par ses propriétés, par ses caractéristiques, c'est-à-dire non pas seulement par l'apparence qu'il nous présente mais par les mécanismes et les contraintes qui font qu'il nous apparaît sous cette forme. Ainsi, nous nous plaçons à un plus haut niveau d'abstraction»*.

C'est dans cette optique que des recherches actives ont été entreprises à Nantes dès le début des années 1990 à travers un grand projet nommé «ExploFormes» [Desmontils, 1995b, Lucas et Desmontils, 1995] qui se compose de plusieurs projets autour de la modélisation déclarative :

- PolyFormes [Martin and Martin, 1993, Martin and Martin, 1999] : construction de polyèdres réguliers et semi-réguliers (Fig. 1.12) ;
- MultiFormes [Boughanem et Plemenos, 1994, Ruchaud, 2001] : création de scènes hiérarchiques [Ruchaud and Plemenos, 2002] (Fig. 1.14, 1.15) ;
- SpatioFormes [Poulet, 1994] : constructions à base de voxels ;
- FiloFormes [Pajot-Duval, 1994] : configurations de segments de droite : jeux d'essais et création de tableaux de fils tendus ;
- VoluFormes [Chauvat, 1994] : contrôle spatial et mécanismes de croissance (Fig. 1.13).

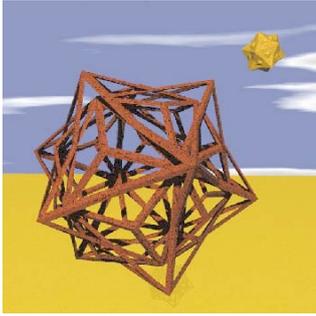


Figure 1.12. PolyFormes

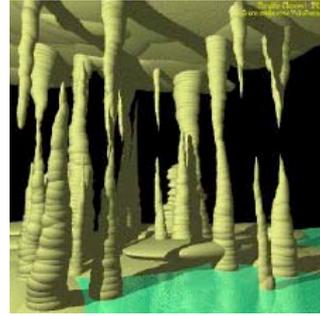


Figure 1.13. VoluFormes

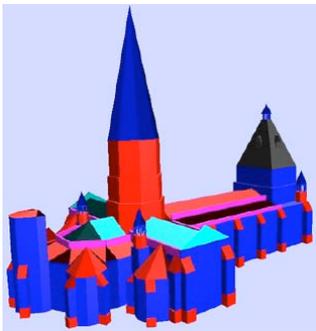


Figure 1.14. MultiFormes : La collégiale du Dorat



Figure 1.15. MultiFormes : La collégiale du Dorat texturée

D'autres projets comme PlaquoFormes [Bouvet, 1988], PastoFormes [Colin, 1990b], AutoFormes [Martin, 1990], UrbaFormes [Le Goff, 1990], MégaFormes [Lucas et Poulet, 1994], CordiFormes [Desmontils, 1998] ou encore CurviFormes [Demoisson, 1994, Robillard, 1996, Daniel, 1996, Rossignol, 2001] ont aussi été menés à l'université de Nantes ainsi qu'à l'École des Mines de Nantes avec BâtiMan [Champciaux, 1998].

À la même époque, en 1996, l'équipe du GRail⁴ à l'université de Washington se penche sur un langage déclaratif adapté au domaine du cinéma et portant sur le contrôle de caméras [Christianson and al., 1996] dans les travelling et les trajectoires de prises de vues.

C'est aussi dès le début des années 1990 que l'on commence à l'IRIT⁵ à s'intéresser de très près à la modélisation déclarative [Djedi, 1991] avec le projet NOVAC [Caubet et al., 1992] destiné aux non-voyants. Une forte collaboration entre plusieurs équipes de ce laboratoire travaillant autour des interfaces Homme-Machine a permis de développer les premières versions de DEM²ONS⁶ [Gaildrat et al., 1993, Kwaiter, 1998, Le Roux, 2003, Le Roux and al., 2004]. DEM²ONS est un outil de conception permettant d'assister l'utilisateur, à travers une interaction multimodale et une approche déclarative, dans le but d'obtenir une scène tridimen-

⁴Graphics And Imaging Laboratory

⁵Institut de Recherche en Informatique de Toulouse

⁶Declarative Multimodal MOdelliNg System

sionnelle complexe. D'autres travaux sont aussi actuellement menés au Laboratoire MSI⁷ de Limoge sur la modélisation déclarative d'ambiance brumeuse [Jolivet and al., 2004].

Depuis ses balbutiements à la fin des années 1980, la modélisation déclarative n'a cessé d'évoluer [Bonnefoi and al., 2004], et le lecteur pourra se rapporter à la thèse l'Emmanuel Desmontils [Desmontils, 1998] et à l'habilitation à diriger des recherches de Véronique Gaildrat [Gaildrat, 2003] pour un tour d'horizon plus complet du domaine de la modélisation déclarative.

1.4.2 Les fondements de la modélisation déclarative

Fort de plusieurs années d'expérience issues du projet *ExploFormes*, il émerge dès le milieu des années 1990 [Lucas, 1993, Desmontils, 1995b, Lucas et Desmontils, 1995] que les modeleurs déclaratifs répondent à un schéma de conception en trois étapes :

- description de la scène ;
- recherche des solutions ;
- prise de connaissance de ces solutions.

En 1997, un document fait le point sur les attentes et sur la vision qu'a un utilisateur d'un modeleur déclaratif [Colin and al., 1997]. Les trois phases y sont rapidement décrites et les auteurs insistent plus particulièrement sur l'aspect méthodologique de la conception attaché au concept du déclaratif. Ils choisissent pour cela de détailler un procédé de construction en spirale (Fig. 1.16) convergeant vers la ou les solutions générées issues d'une description qui se raffine à chaque étape.

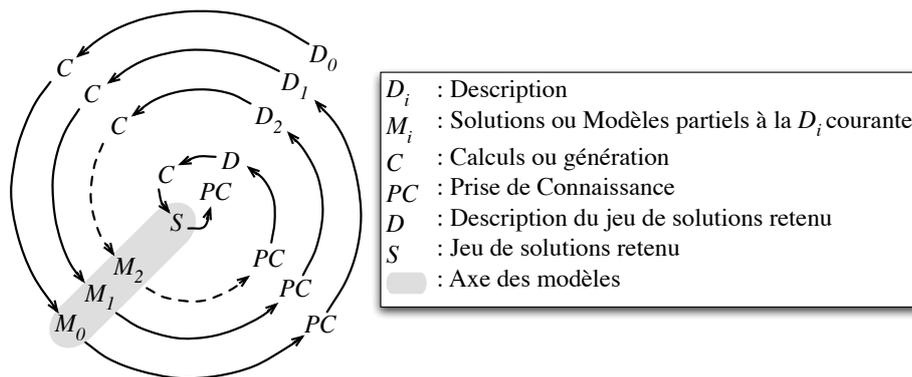


Figure 1.16. Processus de conception en spirale [Colin and al., 1997]

⁷Méthodes et Structures Informatique

1.4.2.1 Description de la scène

La description est le point d'entrée d'un modèleur déclaratif. C'est ce que donne l'utilisateur à la machine pour que celle-ci lui fournisse en retour un ensemble de solutions répondant à sa demande. Les données d'entrée peuvent être de plusieurs natures :

- texte sous forme de langage naturel ou pseudo-naturel ;
- script de description ;
- image ou croquis ;
- menu déroulant et champ de saisie ;
- son ou texte sous forme de langage parlé ;
- *cave*⁸ de réalité augmentée avec capteurs de mouvements . . .

Il est cependant essentiel de travailler avec une description de haut niveau proche de l'attente et des besoins de l'utilisateur. Il n'y a pas, dans l'absolu, un mode de description qui se détache fondamentalement des autres. Plusieurs modes peuvent cohabiter et même être complémentaires dans certains cas.

La phase de description permet de donner les informations nécessaires à la génération des formes attendues. Il est parfois déjà possible, à ce niveau, de vérifier une certaine cohérence des informations entrées dans le système. Dans le cas d'une description erronée ou incomplète, l'utilisateur peut se voir proposer de modifier sa description avant de poursuivre. Notons cependant que s'il est déjà possible de tester certaines contradictions, incohérences ou ambiguïtés (dites de premier type) à cette étape, de nombreuses autres ne se révéleront que lors de la phase de recherche des solutions.

1.4.2.2 Recherche des solutions : génération

La phase de recherche des solutions permet de construire toutes les formes répondant à la description de la phase précédente. On distingue généralement deux types d'implémentation du moteur de génération : l'une est purement *séquentielle* et l'autre est dite *concurrente*. Dans le cas *séquentiel*, le modèleur déclaratif ne peut pas commencer les calculs avant que toute la description ne soit connue. Les résultats ne sont alors présentés à l'utilisateur qu'à la fin de la génération de toutes les solutions. Ce procédé qui a l'avantage d'être simple peut parfois se révéler extrêmement long. L'implémentation *concurrente* [Champciaux, 1997], quant à elle, permet au modèleur de commencer les calculs dès que suffisamment de données sont disponibles. L'utilisateur peut ainsi très vite consulter les premiers résultats.

Cependant, pour être véritablement efficace, le moteur de génération des solutions doit être dédié aux formes qu'il permet d'engendrer. En effet, un tel moteur s'apparente le plus souvent à un système de contraintes à résoudre où chaque contrainte est étroitement liée au type d'objet ou de forme à manipuler. De plus, le passage de la description aux contraintes est d'autant plus précis que le vocabulaire est adapté au domaine de travail.

Après la génération de la ou des solutions, celles-ci sont présentées à l'utilisateur lors de la phase de prise de connaissance des solutions.

⁸On trouve parfois ce terme traduit par «cage»

1.4.2.3 Prise de connaissance des solutions

La dernière phase d'un cycle déclaratif est la prise de connaissance des solutions. Elle permet de montrer à l'utilisateur toutes les formes engendrées par le générateur de la phase précédente. Cette étape est complexe cachant de nombreuses difficultés. En effet il est courant que l'utilisateur donne une description incomplète, ce qui, dans la majorité des cas, va induire plusieurs solutions et parfois même un très grand nombre de solutions. Il est alors très difficile de parcourir cet espace sans être guidé. C'est pourquoi les premiers outils que doit proposer un modeleur déclaratif dans sa phase de prise de connaissance sont des outils de navigation dans l'espace des solutions. Ce problème d'exploration est bien plus complexe lorsque le domaine de solutions n'est pas discret mais continu, ce qui est le cas des surfaces. Lors de cette navigation, les formes résultats doivent être présentées à l'utilisateur, ainsi la visualisation est au centre de la problématique d'exploration des solutions. L'utilisateur doit pouvoir rapidement et facilement distinguer les solutions afin de faire son choix. C'est dans cette optique que Christian Colin dans [Colin, 1990a] propose une méthode de calcul automatique d'une bonne vue d'une scène.

Le ou les résultats ayant été portés à la connaissance de l'utilisateur, celui-ci doit pouvoir valider ou modifier sa description afin d'obtenir celui ou ceux qu'il souhaite. Les outils de modification se dissocient en deux classes, l'une permettant de raffiner et de préciser la description, l'autre autorisant les retouches de la description tout en contrôlant et en restant dans le respect des contraintes établies.

Vient ensuite l'aspect compréhension des résultats, il faut que l'utilisateur puisse rapidement comprendre et connaître les propriétés des formes obtenues. Ainsi un ensemble d'outils de compréhension de la scène sont à intégrer dans un modeleur déclaratif. En effet, si l'utilisateur connaît la forme qu'il veut obtenir puisqu'il en donne une description, il ne connaît pas pour autant toutes les propriétés des formes répondant à sa description. Il est alors possible d'obtenir d'autres formes complètement différentes de celles a priori attendues alors qu'elles respectent toutes les mêmes contraintes. Afin que l'utilisateur puisse s'y retrouver et comprendre un tel comportement, des outils de classement, de mise en évidence des propriétés et même de comparaison sont essentiels dans un modeleur déclaratif.

Tous ces outils, proposés à l'utilisateur afin de choisir la solution qu'il désire, peuvent utiliser la connaissance issue de la description de l'objet pour optimiser les traitements et ainsi parcourir de façon plus efficace l'espace des solutions.

1.4.2.4 Modeleur traditionnel vs Modeleur déclaratif

Le procédé de modélisation part d'une idée quel que soit le modeleur utilisé. Cependant, comme le montre la figure 1.17, les cheminements possibles diffèrent en fonction du type de modeleur utilisé. Un modeleur déclaratif décharge au maximum l'utilisateur de manipulations techniques alors qu'un modeleur traditionnel (ou impératif selon la littérature [Gaildrat, 2003]) impose une implication plus soutenue de l'utilisateur.

Ainsi, lors de l'utilisation d'un modeleur impératif, l'utilisateur doit bien connaître l'outil

qu'il manipule. Il doit être capable de découper en un ensemble d'opérations disponibles dans le modèleur l'objet mental qu'il désire obtenir. La forme réalisée à l'issue de ces étapes de construction fait ensuite l'objet d'un ensemble de tests de validité qui permettent de vérifier les propriétés attendues ou au contraire de déceler les contraintes non vérifiées. Dans le cas de tests négatifs, l'utilisateur doit revoir sa vision de l'objet et en modifier sa construction.

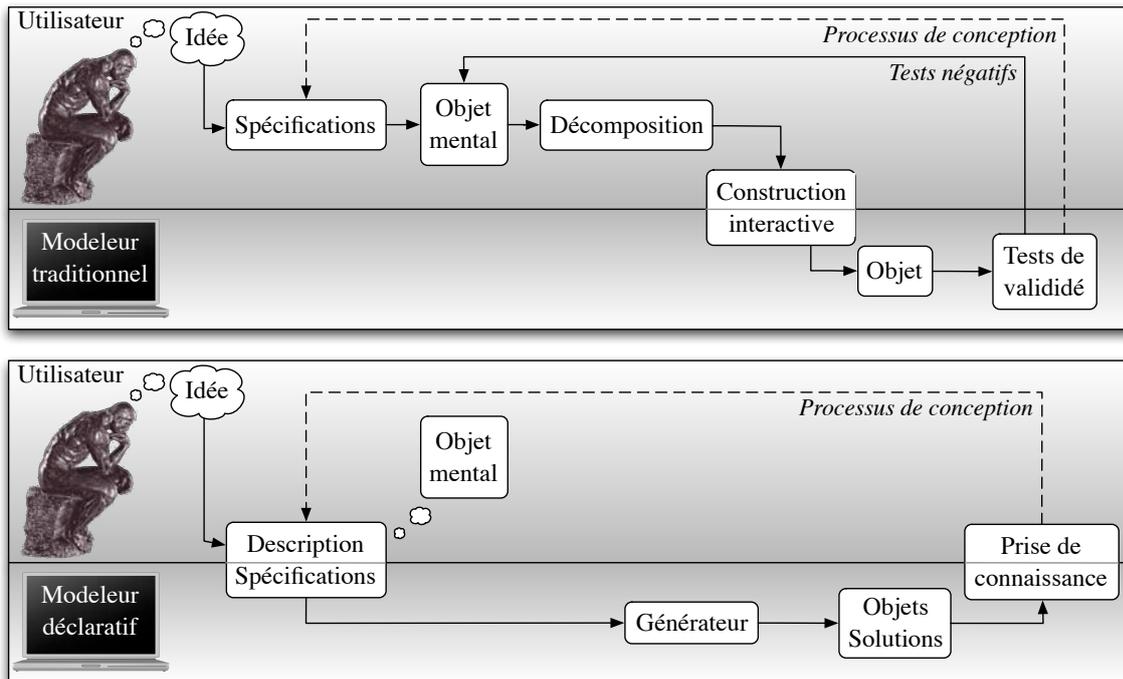


Figure 1.17. Comparaison de modèleurs : traditionnel vs déclaratif [Colin and al., 1997]

La manipulation d'un modèleur déclaratif a une vocation bien différente. L'utilisateur est libéré de la manipulation directe du modèle géométrique, il n'a donc pas besoin d'avoir de pré-requis basés sur la manipulation avancée du modèleur. Contrairement au principe de modélisation par modèleur traditionnel, l'approche déclarative n'utilise pas directement l'objet mental pour la construction, celui-ci reste une image que se fait l'utilisateur du résultat attendu sans vraiment s'en servir de modèle de construction. Un modèleur déclaratif n'a donc besoin que d'une description et éventuellement d'un ensemble de spécifications pour que le générateur puisse en déduire le ou les objets résultats. L'étape de tests et de validité se transforme ici en une étape de prise de connaissance des solutions. Les résultats ayant été générés automatiquement, ils respectent forcément les spécifications de l'utilisateur qui n'a plus qu'à choisir la forme la plus proche de l'objet mental qu'il désire en utilisant un ensemble d'outils de navigation et de compréhension adaptés. S'il le désire, il lui est possible de revenir sur la description faite de l'objet afin de la compléter, de la raffiner ou de la modifier jusqu'à l'obtention de l'objet désiré.

1.4.3 Problématique et besoins de la modélisation de surfaces dans une optique déclarative

Les surfaces à pôles sont les modèles géométriques les plus utilisés dans le monde de la CAO. Or les outils actuels mis à la disposition des concepteurs restent encore très près des modèles mathématiques. Ainsi, la modélisation de surfaces à pôles consiste à positionner dans l'espace un ensemble de points de contrôle afin de respecter un ensemble de contraintes, pour la plupart fonctionnelles [Pace, 1996], tout en gardant une touche artistique forte. Ce travail appliqué à des objets ou des formes complexes ou de grande taille implique la manipulation d'un grand nombre de points de contrôle ce qui constitue un travail long, fastidieux et difficile. L'utilisateur perd alors beaucoup de temps dans la manipulation des outils de conception au lieu de se focaliser sur l'aspect créatif lié à son métier qui est plus de son ressort. C'est alors dans l'optique de la réalisation d'outils de haut niveau permettant de faciliter le travail des concepteurs que s'inscrit l'approche déclarative de la modélisation de surfaces.

Par cette approche, le concepteur donne au modeleur déclaratif de surfaces l'ensemble des spécifications que doit respecter la surface désirée. Le modeleur génère une ou un ensemble de surfaces respectant les propriétés et les contraintes spécifiées. Il ne reste au concepteur qu'à choisir le modèle qui l'intéresse. Cette solution choisie, n'étant parfois qu'une ébauche de la forme souhaitée par le concepteur, pourra être le point de départ de son travail. Il pourra ainsi la modifier jusqu'à obtenir sa réalisation finale. Mais cette ébauche ne renferme pas seulement les données numériques permettant de construire son modèle géométrique, elle est aussi riche de la sémantique qui a permis de la générer.

Cette sémantique peut être utilisée par les systèmes qui seront amenés à manipuler le modèle géométrique afin de réaliser des traitements plus efficaces en utilisant des algorithmes plus adaptés et tenant compte de certaines propriétés de la forme. Cette connaissance de la forme peut aussi être utilisée pour guider et contrôler les modifications apportées par le concepteur tout en gardant la cohérence des contraintes de construction déduites de la description. Ce respect de cohérence et de contraintes lors de modifications est un problème difficile sur lequel peu de réponses ont été apportées. Plusieurs orientations sont envisageables, mais encore à développer. La première consiste à n'autoriser que le mouvement des pôles qui gardent le respect des contraintes du modèle. Une seconde méthode est de demander, en cas de conflit entre plusieurs contraintes, qu'elles sont celles à garder valides, les autres étant alors relaxées. Une troisième méthode consiste à pondérer les contraintes en fonction de l'importance que leur porte l'utilisateur. En cas de conflit inter-contraintes, ce sont celles dont le poids est le plus faible qui sont relaxées en priorité pour maximiser le poids total des contraintes respectées. Ainsi les modèles résultats peuvent être classés par leur pourcentage de respect des contraintes.

Outre tous ces avantages, l'approche déclarative de la modélisation de surface donne au concepteur la possibilité d'obtenir des ébauches de surfaces solutions originales. Respectant

l'ensemble des spécifications initialement données dans la description, ces surfaces peuvent être des solutions au problème du concepteur auxquelles il n'aurait pas forcément pensé. Ce phénomène peut alors stimuler sa créativité et engendrer des surfaces résultats innovantes. Enfin, bien que ce ne soit pas son but premier, de part sa simplicité d'utilisation l'approche déclarative de la modélisation de surfaces permet à des personnes non initiées de construire et produire des formes surfaciques. C'est alors le modelleur qui vient s'ajuster au métier et non le concepteur qui doit orienter ces créations autour des outils disponibles.

Ce paragraphe expose les objectifs à long terme de la modélisation déclarative de surfaces. Il existe deux approches pour y parvenir, celles-ci sont présentées dans la partie suivante.

1.4.4 Approches possibles de la modélisation déclarative de surfaces

Face aux besoins cités dans la partie précédente, il est possible de représenter sous la forme d'un graphique de forme pyramidale (Fig. 1.18) l'association entre les niveaux de description et les outils de manipulations associés. Ainsi trois niveaux de description se détachent, partant du plus bas (niveau 1) qui correspond à la manipulation des pôles de la surface à produire jusqu'au niveau de description le plus élevé (niveau 3) qui correspond aux données d'un ensemble de caractéristiques et de propriétés.

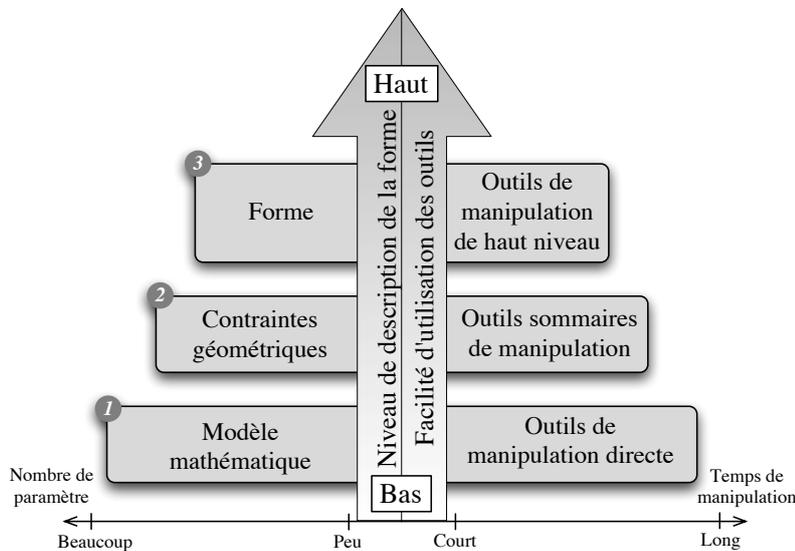


Figure 1.18. Niveaux de description et techniques associées [Pernot, 2004]

Le niveau 2 fait la jonction entre le niveau dédié à l'utilisateur (niveau 3) et le niveau dédié à la machine (niveau 1). Il existe alors deux approches possibles à la résolution de ce problème. La première fait actuellement l'objet de recherches actives au laboratoire 3S⁹ de Grenoble sous la direction du professeur Jean-Claude Léon [Guillet, 1999, Lesage, 2002, Pernot, 2004, Cheutet and al., 2004]. Elle consiste à partir du niveau 1 pour s'approcher du niveau 3 en créant des outils d'un niveau toujours plus élevé et plus proche des besoins

⁹Sols, Solides, Structures

de l'utilisateur, la flèche verticale de la figure 1.18 représente cette progression ascendante. Nous parlons plus en détails des techniques utilisées dans ces recherches à la partie 4.2.2. La seconde approche possible va à contre-sens de la première puisqu'elle consiste à partir des besoins de l'utilisateur (niveau 3) pour se rapprocher des modèles mathématiques (niveau 1). Inévitablement les deux approches sont menées à se rejoindre permettant ainsi de produire un outil puissant et proche des demandes de l'utilisateur, destiné à la modélisation de surfaces.

Dans le cadre de cette étude, nous avons choisi d'exploiter l'approche numéro 2, dite descendante. Le chapitre suivant présente la démarche et l'architecture que nous avons mises en œuvre. Elles seront ensuite détaillées dans toute la suite de ce document.

1.5 Détails de l'approche proposée

1.5.1 La méthode

En s'inspirant de la méthode MERISE¹⁰, la démarche entreprise pour cette étude part de l'analyse des besoins des utilisateurs pour aboutir à une maquette de modèle déclaratif de surfaces. Le problème se subdivise ainsi en plusieurs niveaux d'abstraction (Fig. 1.19), où chaque niveau subit des traitements avant de pouvoir évoluer au niveau inférieur.

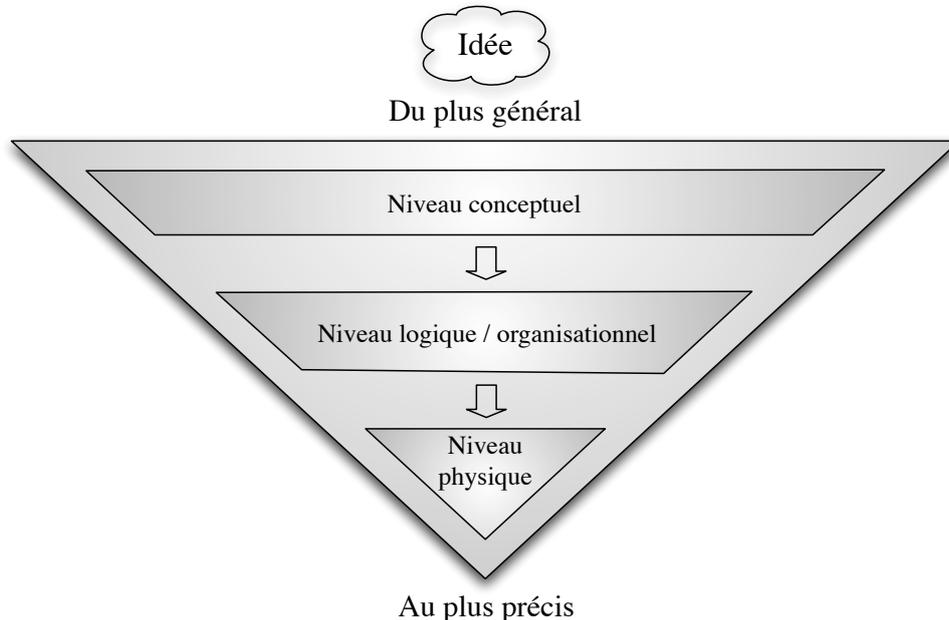


Figure 1.19. Niveaux d'abstraction de la méthode MERISE

¹⁰Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise [Tardieu et al., 1983, Tardieu et al., 1985, Rochfeld et Moréjon, 1986, Tardieu et al., 2001]

Dès qu'une idée apparaît, la méthode¹¹ de conception utilisée prévoit de la modéliser selon trois niveaux de détails (ou d'abstraction) :

- **Niveau conceptuel** : C'est le niveau le plus haut, la question n'est abordé que par le «Quoi» et le «Pourquoi» . Il n'est pas ici question de rentrer dans le détail, il s'agit simplement de présenter les besoins ;
- **Niveau logique/Organisationnel** : C'est une précision du niveau conceptuel en détaillant «Comment» sont réalisées les tâches et comment sont structurées les informations ;
- **Niveau physique** : C'est le plus bas niveau d'abstraction, le fonctionnement matériel et les détails d'implémentation y sont expliqués.

1.5.2 L'idée

Utilisant cette méthode et s'inspirant des travaux réalisés au cours du projet *Explo-Formes*, le premier niveau d'abstraction proposé pour le modeleur déclaratif de surface se compose de trois grandes étapes (Fig. 1.20) :

- Description ;
- Génération ;
- Prise de connaissance des solutions.

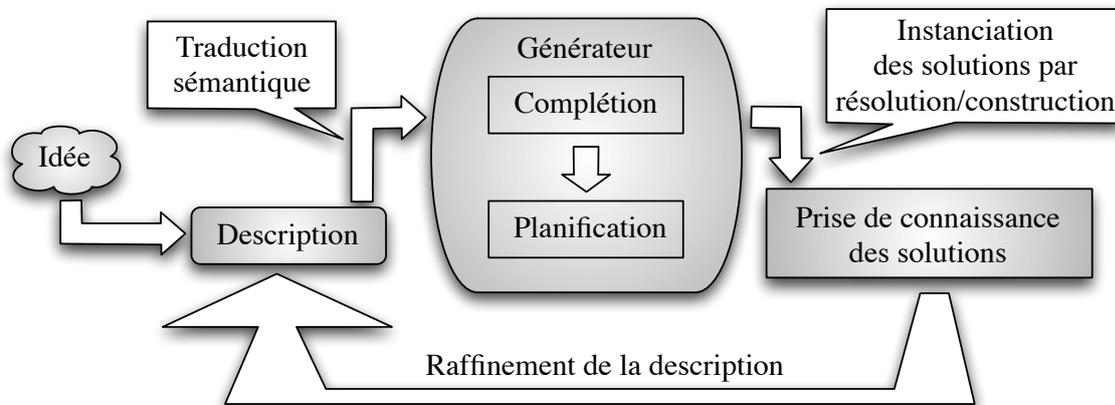


Figure 1.20. Processus déclaratif général proposé

L'idée est donnée par l'utilisateur sous la forme d'une description [La Greca et al., 2002] au modeleur déclaratif. Les informations nécessaires sont envoyées au *générateur* qui a deux fonctions principales : compléter la description si nécessaire et résoudre le problème. Le ou les résultats sont ensuite présentés à l'utilisateur qui peut accepter une ou plusieurs solutions ou modifier sa description et relancer le processus pour obtenir de nouveaux résultats.

¹¹Une «méthode» est dite *opérationnelle* si elle regroupe une démarche, un formalisme et des moyens de mise en œuvre.

1.5.3 L'approche conceptuelle

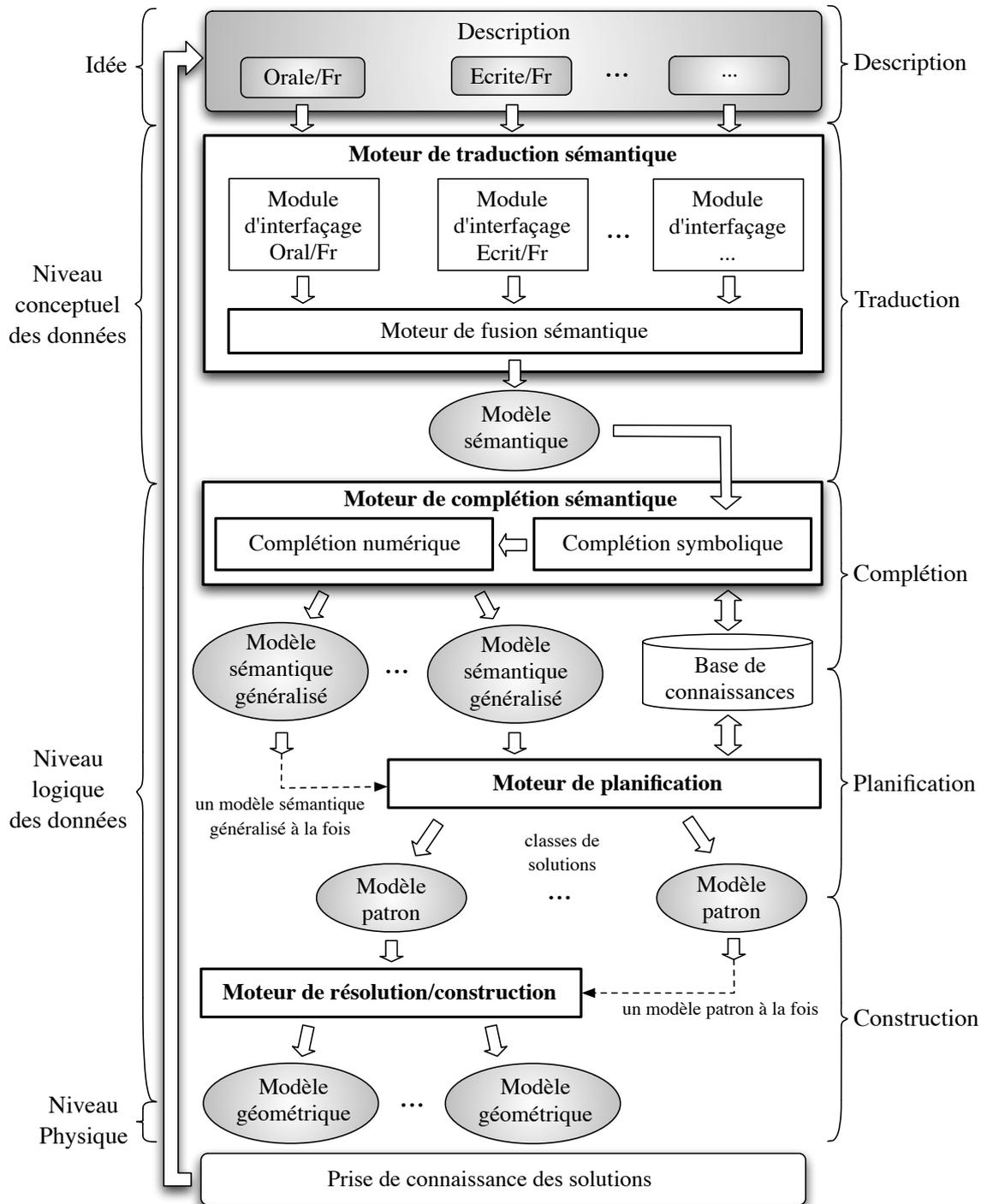


Figure 1.21. Architecture du processus déclaratif proposé

Induite par la méthode utilisée, l'architecture de la maquette du modèleur déclaratif ici développé suit un schéma d'abstraction similaire [La Greca et Daniel, 2002]. Dans le détail de l'architecture du processus déclaratif proposé (Fig. 1.21), il apparaît clairement que la description est la représentation de plus haut niveau des modèles solutions alors que l'implémentation des modèles géométriques en est la représentation de plus bas niveau.

Le travail réalisé se porte tout d'abord sur l'information de plus haut niveau disponible. Cette information n'est autre que la description de la surface que veut obtenir l'utilisateur. Celle-ci est assimilable à une «idée» en terme de niveaux d'abstraction. C'est dans le chapitre 2 que l'étude de faisabilité concernant la *description* de surfaces est détaillée en répondant aux questions : «Est-il possible de décrire une surface ? Et sous quelles conditions ?».

Cette étude [Parisot et al., 2002] se révélant positive, l'analyse se poursuit au niveau *conceptuel*. Ce travail révèle un fort besoin de structuration de l'information afin qu'elle soit exploitable par un système informatique. Le concept de «modules d'interfaçages» émerge alors de la réflexion et permet ainsi au processus déclaratif d'être évolutif et ouvert à tout nouveau support de description envisagé par l'utilisateur. Chaque module prend en entrée un type de description qui lui est propre, comme par exemple une description sous forme de texte écrit en français ou bien d'un discours oral en anglais, et en génère un modèle *sémantique* intégrant l'information de l'utilisateur de manière structurée selon un langage *sémantique* de description fixé. Lorsque plusieurs descriptions d'un même objet sont fournies sous plusieurs formes, le **moteur de fusion sémantique** intègre tous les modèles *sémantiques* associés pour n'en générer plus qu'un nommé *modèle sémantique* de la surface.

La description donnée par l'utilisateur étant rarement complète, le modèle *sémantique* de la surface est injecté dans le **moteur de complétion sémantique** du générateur. Celui-ci *interprète* le modèle *sémantique* afin de générer un ou plusieurs *modèles sémantiques généralisés* enrichis d'un ensemble de données déduites de la base de connaissances. Pour chaque *modèle sémantique généralisé* ainsi créé, le **moteur de planification de construction** du générateur déduit un ensemble de classes solutions où une classe de solutions est aussi appelée *modèle patron* [La Greca and Daniel, 2004a]. Les modèles *sémantiques*, *sémantiques généralisés* et *patrons* utilisés dans notre approche ainsi que les processus de complétion *sémantique* et de planification qui y sont rattachés, sont détaillés dans le chapitre 3.

Par la définition même de classe, chaque modèle *patron* peut définir un ensemble de solutions. Ainsi une solution particulière au problème est une instance d'un modèle *patron*, sa représentation est alors un *modèle géométrique*. Cette instantiation de solution se traduit par la *construction* d'un modèle *géométrique* particulier. Ce processus est effectué

dans le **moteur de résolution/construction** par une méthode de déformation de surface NURBS. Cette méthode permet de résoudre des contraintes géométriques ponctuelles de passage en contrôlant la forme et le comportement de l'influence de ces mêmes contraintes [La Greca et Daniel, 2004, La Greca and Daniel, 2004b, La Greca and al., 2005]. Tous les traitements que nous avons développés et qui sont effectués sur les surfaces NURBS sont expliqués et démontrés dans le chapitre 4.

Enfin, le chapitre 5 montre les résultats obtenus à l'issue de cette étude et présente les applications qui ont été développées tout au long de cette thèse.

1.6 Conclusion

Ce premier chapitre introduit toute l'étude sur l'approche déclarative de surfaces en présentant un rapide état de l'art de la modélisation en CAO et un bref aperçu de la problématique déclarative dans ce domaine. Nous y précisons le choix de l'utilisation de surfaces paramétriques de type NURBS dont nous donnons la définition. Enfin l'approche déclarative de la modélisation de surfaces fait l'objet d'une analyse *conceptuelle* où est introduit l'ensemble des modèles impliqués dans l'approche descendante que nous proposons. L'analyse *logique* du processus déclaratif que nous proposons d'appliquer à la modélisation de surfaces est détaillée dans l'ensemble des chapitres de ce document. Tous les modèles ainsi que tous les processus qui y sont rattachés sont développés et illustrés un à un.

DESCRIPTION DE SURFACES

2.1 Introduction

La description d'objets, de scènes ou de situations a de tout temps existé. Dès la rhétorique ancienne, elle appelle surtout à l'éloge de personnages, de lieux ou de situations particulières. Elle est aussi présente dans la poésie, notamment dans les poèmes homériques où la description met en avant l'aspect précieux des objets comme les casques ou les boucliers. Dans son approche moderne, la description se rattache à un courant artistique nommé le *réalisme* apparu en 1833 qui se retrouve dans la littérature et la peinture. Ce mouvement tend à écarter l'idéalisation des artistes qui se doivent alors de ne pas modifier le réel ou de ne pas volontairement donner des images incomplètes. Jean Cocteau dans son *Essai de critique indirecte* dit que «*Le vrai réalisme consiste à montrer les choses surprenantes que l'habitude cache sous une housse et nous empêche de voir*». Il met dans cette phrase l'accent sur le fait que le *réalisme* est avant tout un retour vers l'attention du *banal*. Attaché à ce mouvement artistique moderne, la description dite *réaliste* connaît son âge d'or dans le roman réaliste de Flaubert à Zola. En 1872, Émile Littré lui donne alors une définition, restée inchangée à ce jour, comme étant un «*discours par lequel on dépeint*» [Littré, 1872]. La *description* d'une chose est ainsi sa représentation par le discours. Ce discours doit donc être assez riche pour pouvoir exprimer et détailler l'allure, la forme, les couleurs, les textures, le positionnement dans l'espace, l'évolution et même l'utilité de l'objet dépeint. Ce chapitre expose les premiers travaux entrepris au cours de cette thèse permettant de décrire des surfaces. Il montre à travers deux expériences que la description de surfaces est possible et en définit les conditions d'étude. Dans une première partie nous présentons les moyens et outils de description existants et pouvant être adaptés à la modélisation de surfaces. Nous nous penchons ensuite, à travers une première expérience, sur l'impact que peut avoir le contexte de travail et l'orientation métier sur une description. En se focalisant sur le contexte métier de la mécanique, dans la troisième partie de ce chapitre, nous identifions et détaillons toutes les catégories de vocabulaire pouvant intervenir dans la modélisation déclarative de surfaces. Enfin, avant de conclure, nous terminons ce chapitre par la présentation des résultats et du protocole de la seconde expérience menée en colla-

boration avec l'équipe IMS¹ du LSIS² et l'ENSAM³ d'Aix-en-Provence. Cette expérience a permis de montrer qu'il est possible de décrire des pièces mécaniques et qu'une description peut engendrer des formes cohérentes et proches du modèle existant.

Tout ce chapitre touche à des domaines qui dépassent largement le cadre, déjà important, de la CAO comme par exemple le traitement de la langue naturelle ou le traitement de la langue écrite et orale. Ces domaines de recherche n'étant pas les nôtres, nous n'avons pas la prétention de proposer des solutions mais plutôt de faire émerger des besoins issus de notre problématique.

2.2 Les moyens de description

«*Comment décrire ?*» Ou plutôt «*Comment faire naître une image, une forme par l'échange d'informations ?*» Tout le monde a déjà été confronté à cette situation en voulant décrire une image, un paysage ou un objet par exemple. L'exercice est d'ailleurs souvent très difficile et la seule utilisation de mots est parfois insuffisante. La problématique que nous abordons dans ce chapitre est encore plus complexe puisque l'échange ne se réalise plus entre deux êtres humains, mais entre un être humain et un système informatique. Il faut donc faire appel à la branche de l'informatique attachée au domaine de l'interface Homme-Machine.

Plusieurs méthodes de description d'objets ou de scènes virtuelles ont été abordées dans la littérature. En voici un bref aperçu :

1. Une méthode consiste à utiliser des systèmes de Lindenmayer ou *L-Systemes*. Inspiré des travaux de N. Chomsky [Chomsky, 1956], la *description*⁴ se présente sous forme d'une grammaire formelle composée de quatre éléments : $G = \{V, S, \omega, P\}$ où
 - V : ensemble de symboles qui peuvent être réécrits (variables) ;
 - S : ensemble de symboles qui ne peuvent pas être réécrits ;
 - ω : chaîne de symboles de V définissant l'état initial du système ;
 - P : ensemble de règles de réécriture permettant de reformuler un élément de V (variable) à partir d'un ensemble d'éléments de V et de S (constantes).

Les L-Systemes peuvent être déterministes ou probabilistes. La génération des objets se fait par l'application itérative des règles du L-Système à partir de l'état initial du système jusqu'à son développement maximal. Cette méthode de description, qui donne des résultats qui sont proches des procédés fractales, pourrait se qualifier de systémique. Elle est principalement utilisée pour la génération automatique de plantes [Prusinkiewicz and Lindenmayer, 1990, Prusinkiewicz and al., 1999] (Fig. 2.1) ou de

¹Ingénierie, Mécanique, Systèmes

²Laboratoire des Sciences de l'Information et des Systèmes

³Ecole Nationale Supérieure d'Arts et Métiers

⁴Le terme de *description* est ici utilisé au sens large puisqu'elle se rapproche plus d'une définition d'un système de construction.

bâtiments [Marvie and al., 2005] (Fig. 2.2). La méthode de description des objets reste cependant difficile et peu intuitive.



Figure 2.1. Arbuste défini par un L-System



Figure 2.2. Bâtiment défini par un FL-System

2. Jean-Philippe Pernot [Pernot, 2004] utilise la terminologie de Leyton [Leyton, 1988, Leyton, 1992, Leyton, 2001] pour décrire des courbes planes. L'allure de la courbe est décrite selon ses critères de courbures (maxima, minima) et leurs variations (points d'inflexion). Cette grammaire de formes utilise M (respectivement m) pour désigner les maxima (respectivement minima) locaux de courbure suivis de leur signe et 0 pour les points d'inflexion. Ainsi la courbe fermée de la Figure 2.3 peut être décrite par « $M^+0m^-M^-m^-0M^+m^+M^-m^+$ » :

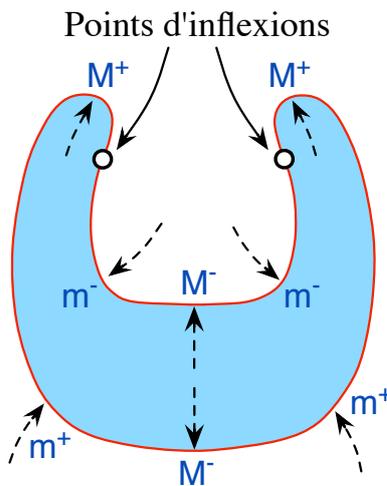


Figure 2.3. Courbe fermée décrite par « $M^+0m^-M^-m^-0M^+m^+M^-m^+$ » [Pernot, 2004]

Il est important de noter que les extrema définis peuvent être rattachés à une certaine sémantique en fonction du procédé qui a permis de les obtenir :

- M^+ : protrusion
- m^- : échancrure
- m^+ : écrasement
- M^- : résistance interne

Ce langage de description réservé aux courbes planes exploite une propriété importante des objets qu'il décrit et la formalise en une grammaire simple suffisante pour proposer une allure à la forme désirée.

3. Certains logiciels de modélisation ou de synthèse d'images comme *PovRay* par exemple décrivent des objets ou des scènes impliquant plusieurs objets en utilisant des scripts. Ces langages de description peuvent être assimilés à de la programmation où tous les éléments sont décrits selon des coordonnées et des primitives d'affichage. Sur la même idée et afin de promouvoir la navigation dans des environnements virtuels sur le web, le langage *VRML*⁵ permet de décrire des scènes animées ou statiques à partir d'un langage interprété. Cette solution permet de décrire de façon très précise un environnement ou un objet mais est plutôt destinée à un utilisateur confirmé et initié au domaine de la modélisation et de la programmation.

Mais le véritable besoin de la modélisation déclarative réside autour d'une interface conviviale et simple d'utilisation. Le cas d'une approche appliquée aux surfaces n'échappe pas à la règle. Il faut que l'utilisateur puisse d'une manière «naturelle» ajouter ou modifier des propriétés ou des caractéristiques à la surface qu'il souhaite obtenir. Comme nous l'avons déjà dit dans la présentation générale de la modélisation déclarative (Cf. 1.4.2.1), plusieurs méthodes sont envisageables à travers :

- une interface graphique appropriée intégrant un ensemble d'outils ciblés tels que des menus déroulants, des boutons, des champs de saisie . . . permettant de guider l'utilisateur tout en lui permettant un très grand éventail de possibilités ;
- des scripts de description ou des grammaires proches de ceux et celles décrits plus haut afin de garder certains outils maîtrisés par certains spécialistes ;
- l'immersion dans une *cave* de réalité augmentée [Cruz-Neira and al., 1993] où l'utilisateur peut directement manipuler la surface avec des outils virtuels du domaine métier choisi [Cutler and al., 1997] ;
- des croquis [Fraczak, 1994, Huot et al., 2004, Alexe and al., 2004]. L'utilisateur dessine une ébauche de la forme qu'il souhaite obtenir sur une tablette graphique, sur un écran tactile ou sur une simple feuille de papier qui sera ensuite numérisée ;
- des sons ou une description orale dans la langue naturelle ou pseudo-naturelle de l'utilisateur [De Mori, 1998] ;
- un texte écrit dans la langue naturelle ou pseudo-naturelle de l'utilisateur.

⁵ *Virtual Reality Markup Language* ou *Virtual Reality Modeling Language*

L'idéal restant la réalisation d'un système multimodal [Dragicevic, 2004, Huot and al., 2004] qui puisse véritablement prendre en compte tous ces moyens d'interaction. L'intégration de toutes ces informations provenant de capteurs différents donne à la description une dimension plus proche de celle de l'utilisateur.

Cette description doit cependant pouvoir laisser à l'utilisateur la possibilité de donner et d'exprimer des propriétés vagues et imprécises ce qui reste l'un des points forts de la modélisation déclarative.

Enfin, le procédé de description peut se réaliser en une seule fois, l'utilisateur décrivant complètement la surface qu'il souhaite obtenir, ou bien de façon progressive permettant à l'utilisateur de voir les étapes successives de construction.

Toutes ces spécifications restent cependant un idéal à long terme qui doit être appliqué à la modélisation déclarative de surfaces. Dans le cadre de cette thèse la problématique de la description a été abordée afin de pouvoir savoir si ce principe est applicable à la modélisation de surfaces. Cette étude a abouti à des pistes de travail basées sur la description à travers un langage pseudo-naturel. Elles seront abordées dans la suite de ce chapitre. L'étude a aussi permis d'aboutir à une architecture faisant intervenir la notion de *modules d'interfaçage* qui sont détaillés dans la partie 3.2.1 de ce document. Néanmoins, quel que soit le moyen donné à l'utilisateur pour s'exprimer, il s'avère que le contexte d'activités et la subjectivité ont une importance et un impact capital sur la description qu'il peut réaliser.

2.3 Importance du contexte et orientation métier

Bien que l'on soit tenté de penser qu'il est possible d'établir une méthode de description générique de surfaces en partant «*du général vers le particulier*», on s'aperçoit très vite que la réalité est toute autre : «*Le procédé de description reçoit une influence cognitive et contextuelle très forte*».

Ainsi la description d'un objet ou d'une forme est différente d'un individu à l'autre en fonction de ses connaissances, de sa subjectivité et du contexte dans lequel il se trouve. Cette constatation avait déjà été faite par Demoisson [Demoisson, 1994] dans le cadre de la description de courbes. Afin de s'assurer que cette réalité soit aussi valable pour les surfaces, nous avons entrepris une rapide expérience se présentant sous la forme d'un sondage dont voici le sujet : «*Décrire la forme représentée sur la figure 2.4*». Cet exercice a été proposé à plusieurs personnes d'horizons différents, et dès les premières descriptions recueillies (voir annexe B.1), il est apparu évident que la description de surfaces suivait les hypothèses de contexte et de subjectivité qui avaient été posées. L'expérience n'a donc pas été poursuivie sur un plus grand nombre de sujets.

Afin de mieux répondre à ce type de comportement, l'idée est de travailler avec un lexique contextuel orienté «*métier*». C'est pourquoi dans tout le reste de cette étude nous nous plaçons dans le contexte métier de la «*mécanique*».

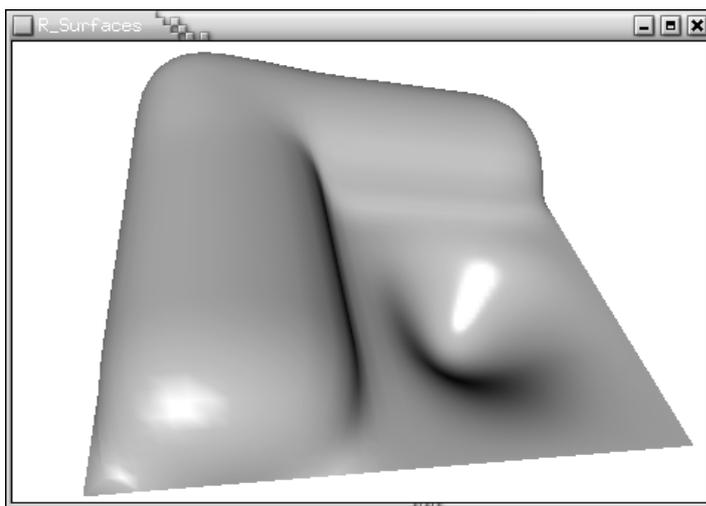


Figure 2.4. Surface à décrire

2.4 Les besoins en vocabulaire pour la description de surfaces

La description en langage naturel ou pseudo-naturel est composée d'un ensemble d'entités élémentaires : les mots (ou lexèmes). Tous ces mots sont regroupés dans un ensemble que l'on nomme *lexique*. Chaque mot de ce lexique a une nature précise qui justifie sa place dans la syntaxe de la description grâce à une *grammaire* et appartient à une catégorie bien précise de vocabulaire.

Demoisson [Demoisson, 1994] construit en 1994 un premier lexique dans le but de décrire des courbes. Pour cela il réalise une enquête à l'École Centrale de Nantes ainsi qu'une étude de faisabilité en collaboration avec l'École des Beaux Arts de Nantes et s'inspire de domaines existants pour trouver du vocabulaire susceptible d'être exploitable. Il a ainsi listé un ensemble de termes dont certains sont issus d'ouvrages traitant de morphopsychologie et de graphologie. À travers son étude de faisabilité, il montre qu'une description est suffisante pour obtenir des modèles de courbes très proches de la courbe désirée. Il est cependant lui aussi confronté au problème de la subjectivité des termes utilisés lorsque les descriptions deviennent de plus en plus imagées.

Toujours dans le domaine de la description de courbes, Vincent Rossignol [Rossignol, 2001] en 2001 enrichit ce lexique en touchant aux domaines de la typographie et de la calligraphie. Jean-Christophe Chambelland fait enfin évoluer ce vocabulaire au domaine des surfaces [Chambelland, 2001] en 2001 et l'utilise, couplé à la notion de grammaire, pour construire un arbre syntaxique de description.

En s'appuyant sur ce travail et en s'inspirant de celui d'A. Pace [Pace, 1996], on peut, a priori, identifier huit catégories (Fig. 2.5) de vocabulaire en vue de décrire des surfaces :

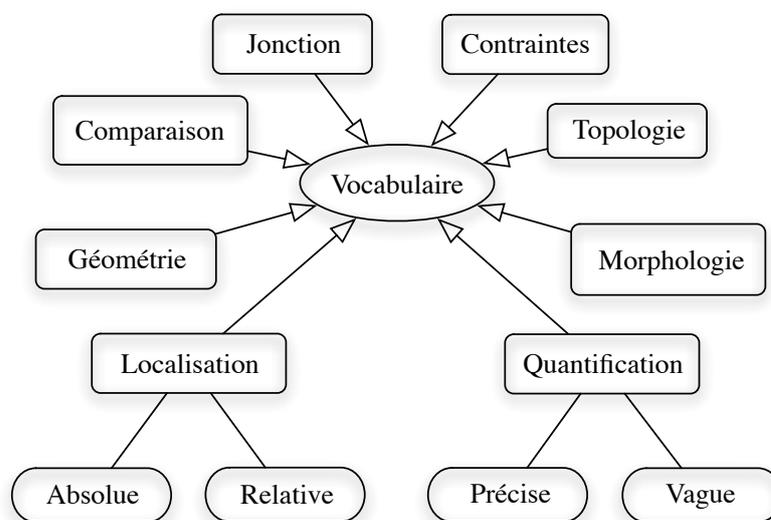


Figure 2.5. Les catégories de vocabulaire

Ce lexique peut aussi s'organiser en trois groupes distincts de vocabulaire (Fig. 2.6) :

- Le *vocabulaire général* qui est tiré de la description visuelle [Petit et Morteau, 1996] (paysage, morphologie, graphologie, géométrie ...);
- Le *vocabulaire «métier»* qui est dépendant du domaine d'application de l'objet à modéliser et du procédé de fabrication;
- Le *vocabulaire personnalisé* qui est le fruit de l'apprentissage du lexique : c'est l'utilisateur lui-même qui définit son propre vocabulaire (par jonction de vocabulaires existants par exemple).

De plus, afin de prendre en compte le niveau de connaissances de l'utilisateur, le vocabulaire général et le vocabulaire «métier» peuvent se diviser en plusieurs parties : *standard* et *expert* par exemple.

Enfin, l'impact de la subjectivité du vocabulaire est fortement impliqué dans la description. Ce phénomène est directement lié aux considérations cognitives de l'utilisateur et peut donc entraîner des ambiguïtés sur un même lexème.

Les différentes catégories de vocabulaire devant apparaître dans le lexique vont maintenant être détaillées. Le vocabulaire associé ne figure cependant qu'en annexe B.2.2 pour ne pas surcharger le texte.

2.4.1 La localisation

Dans sa thèse, Nicolas Denand [Denand, 2004] traite le problème du positionnement dans l'espace en travaillant sur les phrases locatives statiques du français aux niveaux lexical, syntaxique, sémantique et pragmatique. Dans son document, il propose un lexique

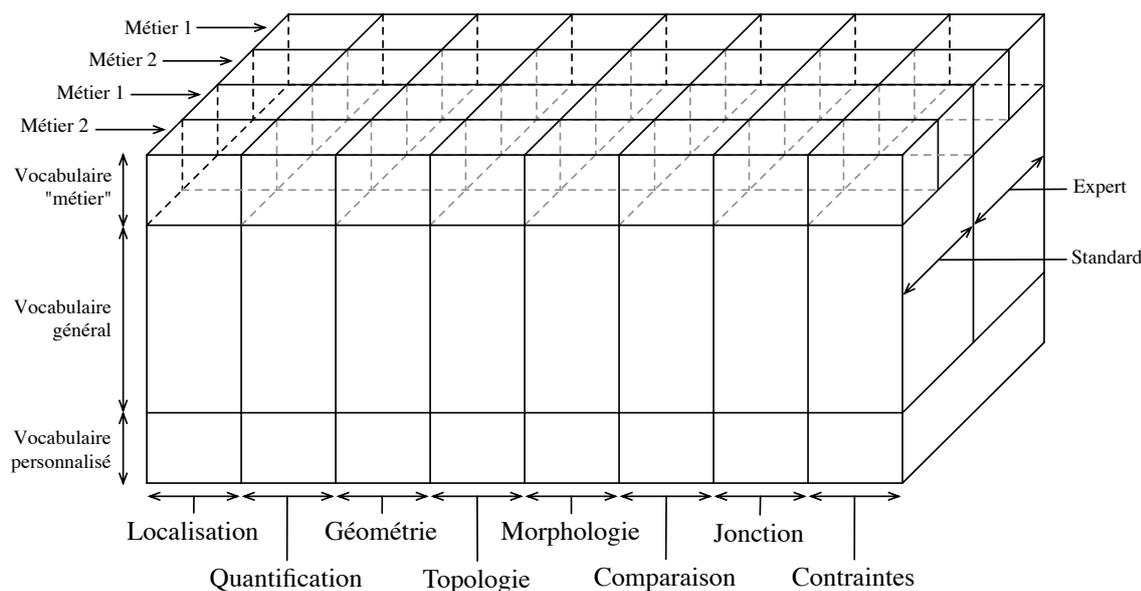


Figure 2.6. Le vocabulaire du lexique

précis dédié à la localisation et introduit une représentation des connaissances adaptée au positionnement. Enfin il identifie plusieurs critères perceptifs et cognitifs permettant d'identifier et de classer les «meilleurs» éléments *référents* d'une scène. Ce qui lui permet, ayant une connaissance locative de tous les éléments d'une scène, de répondre aux questions en *où* de façon très pertinente.

À Paris, l'équipe de recherche de Gérard Ligozat au LIMSI⁶ travaille sur le raisonnement temporel et spatial en se penchant plus particulièrement sur la description d'itinéraires [Gryl and Gonçalves, 1995]. Ces études basées sur la description langagière ont été appliquées à la description de trajets en contexte semi-urbain [Edwards et al., 1995] puis ont été utilisées en corrélation avec des bases de données géographiques [Maya, 2002]. Ce travail a enfin été adapté au guidage de robots par indications [Marciniak and Vetulani, 2002].

Cette catégorie de vocabulaire entre dans sa grande majorité dans la partie du vocabulaire général du lexique. La localisation, ou *positionnement*, n'est pas particulièrement dédiée à un métier ou à une catégorie d'individus mais est clairement impliquée dans tous les problèmes liés à la notion d'espace qu'il soit plan ou volumique.

Dans notre problématique, cette catégorie contient des mots permettant à l'utilisateur de situer une surface dans une scène⁷ et d'en identifier des points ou des zones.

L'idée de localisation implique l'introduction de deux notions très fortement connexes : la position de l'observateur et les éléments référents de la scène. C'est la position de l'observateur qui oriente l'espace de travail. La localisation de tous les éléments impliqués dans

⁶Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur

⁷Espace d'immersion de la surface pouvant contenir plusieurs objets

la description, et présents dans le volume de la scène, est relative à la position de l'utilisateur. Cette position doit être définie au début de la description. En l'absence de précision, l'utilisateur est placé à une position fixe dans le référentiel absolu de la scène (Fig. 2.7).

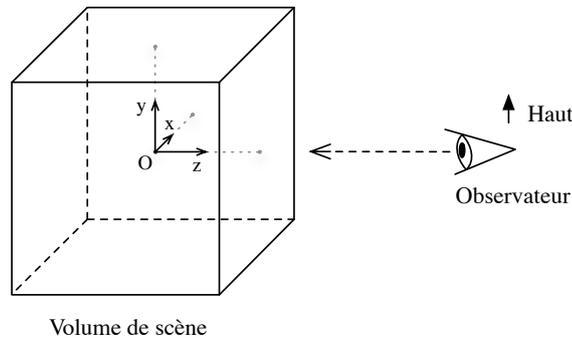


Figure 2.7. Référentiel absolu par défaut du volume d'une scène

Aux éléments référents de la scène s'ajoutent deux niveaux de localisations : la localisation dite *relative* et la localisation dite *absolue*.

2.4.1.1 Éléments référents de la scène

On appelle *éléments référents de la scène* tout objet immergé dans le volume de travail auquel l'utilisateur peut faire appel pour décrire la surface. Lors de la description d'une surface, l'utilisateur doit pouvoir la situer dans son environnement. Cette notion est primordiale pour :

- placer l'objet décrit dans sa scène, ex : «*La surface est au dessus du bloc moteur*» ;
- vérifier certaines contraintes de raccordement à une surface existante de la scène, ex : «*La surface est le prolongement du pare-brise*» ;
- satisfaire les contraintes dites fonctionnelles (Cf. 2.4.7), ex : «*La surface permet une entrée d'air au niveau du filtre à air*».

Les objets de la scène doivent être connus afin de pouvoir leur faire référence lors de la description. L'utilisation d'*alias* vers ces objets particuliers de la description permet une meilleure clarté du discours et évite de nombreuses ambiguïtés. Un *alias* sera aussi nommé *identifiant* dans la partie 2.5.4. Ces objets référents sont de plusieurs ordres :

- un objet matériel de la scène comme «*le bloc moteur*» d'une voiture ;
- un élément immatériel comme le point d'origine *O* du repère de la scène ;
- une partie d'un objet matériel : «*la partie supérieure du bloc moteur*» ;
- une partie de la surface décrite : «*Soit A la zone Haut-droite de la surface*» ;
- une partie du volume de la scène : «*Soit B la partie Avant-Bas-Gauche définie dans l'angle du volume de la scène*».

Les éléments référents sont, la plupart du temps, utilisés dans le cadre de la localisation relative, que nous abordons dans le paragraphe suivant, ou dans celui de la définition de contraintes (Cf. 2.4.7).

2.4.1.2 Localisation relative (ou déictique)

La localisation relative permet de situer un objet ou une partie d'un objet par rapport à un ou plusieurs autres objets. Le vocabulaire de localisation relative dépend donc d'un ou plusieurs éléments auxquels il fait référence. Exemple : «à coté de», «au dessus de» ...

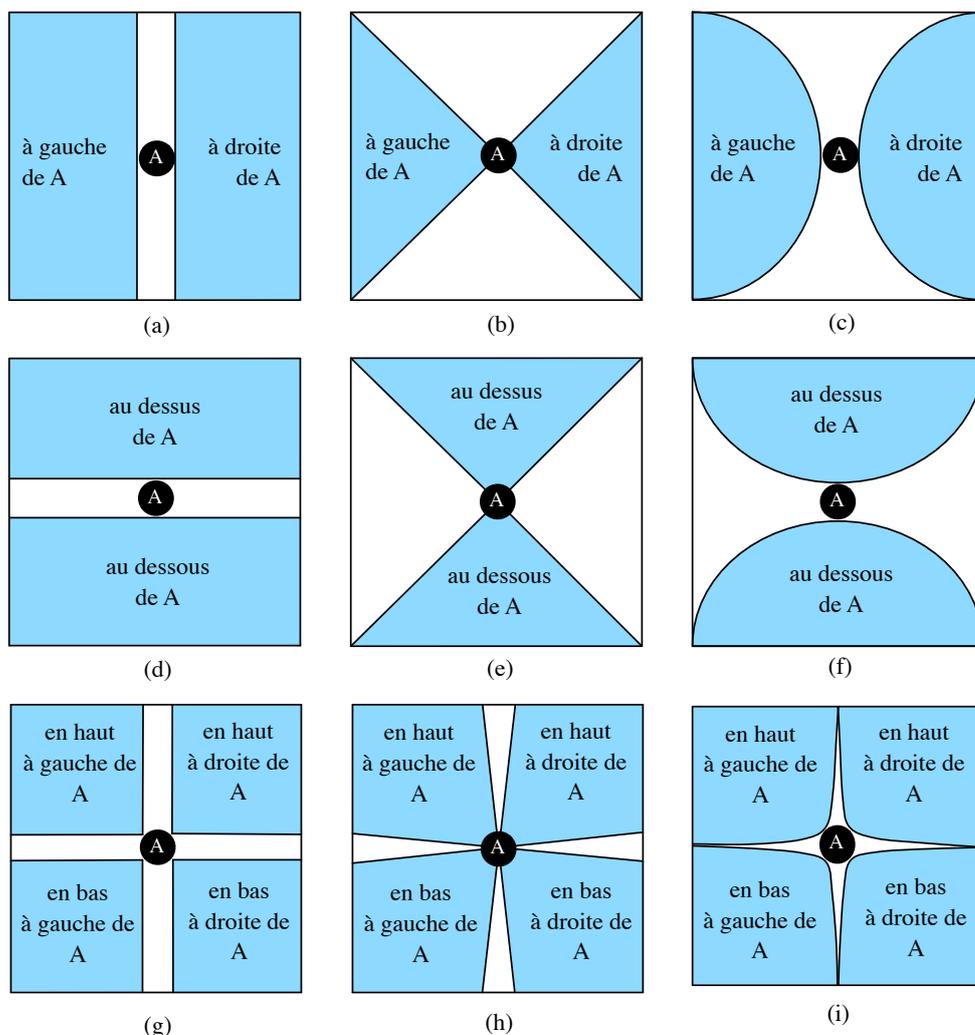


Figure 2.8. Localisations relatives à un objet A dans un plan borné carré

La figure 2.8 présente les localisations relatives usuelles que nous proposons et qui peuvent être appliquées à un objet A dans un plan borné carré. Plusieurs approches sont envisageables pour une même notion et son choix dépend de l'utilisateur et du contexte d'utilisation. Ainsi, dans notre approche, nous faisons intervenir la notion de *configurations* d'une même localisation. Ce sont les différentes façons de représenter une même localisation. Les schémas 2.8.a, 2.8.b et 2.8.c représentent par exemple, la même localisation : «à droite de A », mais cette notion est ici déclinée en trois *configurations* possibles. Nous définissons alors le concept de localisation «ambiguë» lorsqu'une localisation acceptant plusieurs configurations est utilisée sans qu'aucune d'elles ne soit précisée.

La localisation relative par opérateurs flous [Desmontils, 1995a] est aussi envisageable. Le principe consiste à reprendre les zones de la figure 2.8 par exemple en leurs associant des probabilités de présence qui diminuent au fur et à mesure que l'on se rapproche des bords. Cette technique permet, en effet, de favoriser une zone sans vraiment éliminer ses abords. La localisation relative peut aussi faire intervenir du vocabulaire tels que «*B est proche de A*» ou «*B est à proximité de A*». Dans ce cas de localisation relative dite de *proximité*, l'élément *B* est considéré «proche de *A*», «peu proche de *A*» ou bien «très proche de *A*» en fonction de la valeur de la distance⁸ qui les sépare. Cette notion est contrôlée par le quantificateur (Cf. 2.4.8) utilisé dans la description.

En outre, il est important de noter que certains objets d'une scène peuvent avoir une sémantique forte qui permet de localement structurer leur espace d'immersion et donc de modifier les localisations qui leur sont relatives. Si l'on prend par exemple une scène contenant l'objet «*table*», la sémantique de cet objet définit de façon naturelle un *dessus* et un *dessous*, dans un usage traditionnel, sans même que la position de l'observateur ait une influence. On parle alors d'orientation *intrinsèque* de l'objet [Gaildrat, 2003] ou d'objet *polarisant* si son influence s'applique sur d'autres éléments à proximité dans la scène. Le pare-brise d'une voiture, par exemple, permet en général d'orienter les sièges du véhicule.

Mais il est aussi possible de localiser des éléments de la surface ou de la scène sans avoir à s'appuyer sur d'autres éléments déjà définis. C'est la localisation absolue.

2.4.1.3 Localisation absolue

La localisation absolue permet de situer un objet ou une de ses parties en ne dépendant que de l'espace de travail et de la position de l'utilisateur. Les termes typiques du vocabulaire de cette catégorie sont : «*à gauche*», «*au fond*» . . .

Plusieurs types de localisation absolue sont possibles en fonction de la dimension de l'espace d'immersion. Dans notre problématique, ce sont les dimensions 2 et 3 qui sont utilisées. La localisation dans un plan permet d'établir des repères dans le plan paramétrique de la surface et ainsi de se positionner sur la surface, variété de dimension 2.

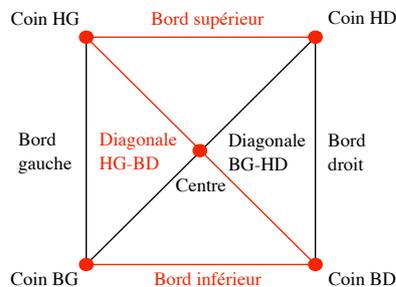


Figure 2.9. Localisations absolues des lignes et points usuels d'un plan borné carré

⁸définie sur l'espace de travail. On considère en général la distance euclidienne qui est plus intuitive à manipuler pour l'utilisateur.

L'observateur étant à la place du lecteur, la figure 2.9 présente les éléments les plus usuels d'un plan borné carré. On y retrouve par exemple ses frontières et ses sommets. Pour identifier le «*coin Haut-Gauche*» il est possible d'utiliser l'abréviation «*coin HG*», il en va de même pour les termes «*Droite*» et «*Bas*». La figure 2.10 présente un vocabulaire permettant de placer les zones usuelles d'un plan borné carré alors que l'observateur est toujours à la place du lecteur.

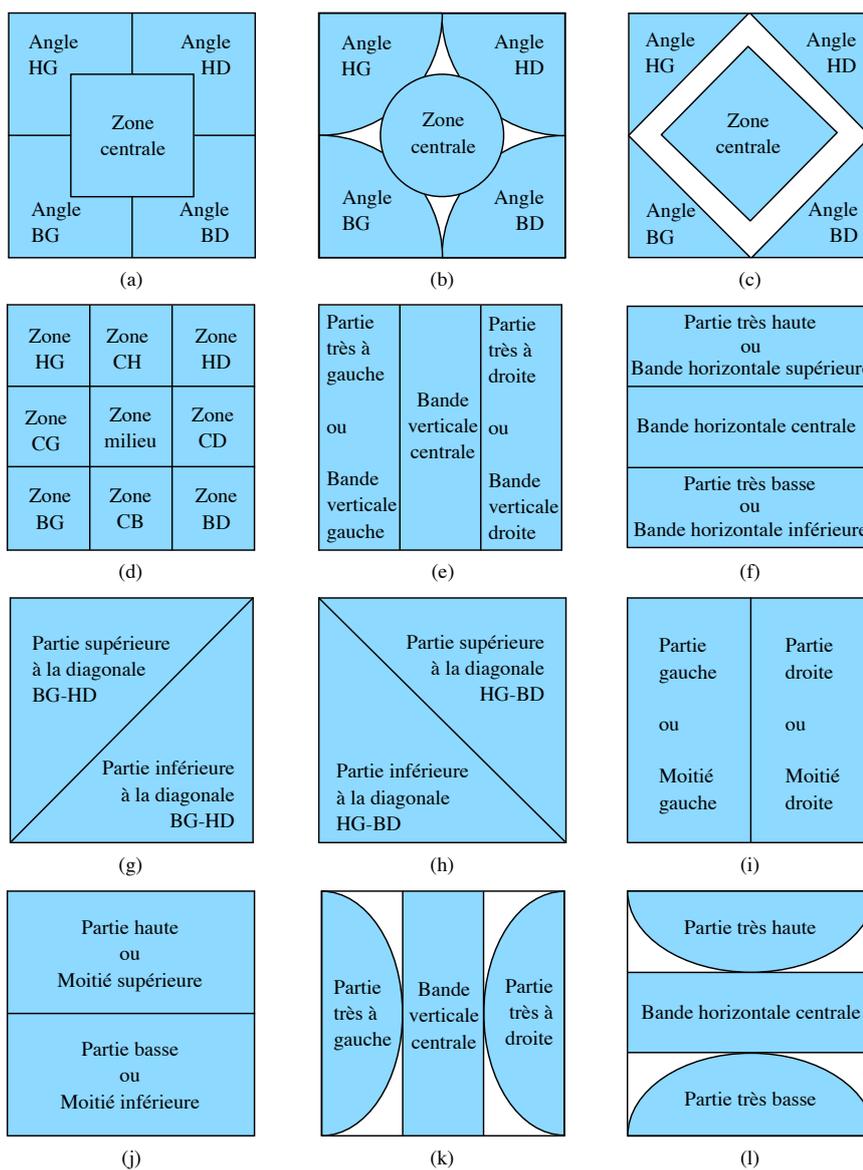


Figure 2.10. Localisations absolues des zones usuelles d'un plan borné carré

L'identification de ces zones, ou parties, usuelles peut se faire avec les mêmes abréviations que précédemment. Leur forme peut varier selon la désignation géométrique faite par l'utilisateur ou en fonction de ses besoins. La notions de *configurations* différentes pour une même localisation est encore ici présente avec les schémas 2.10.a, 2.10.b et 2.10.c pour l'«*angle haut-gauche*» par exemple. On retrouve ainsi le concept de localisation *ambiguë*.

Le passage à la localisation en 3 dimensions se fait de façon assez naturelle et permet une localisation dans une scène, ex : «à gauche», ou dans un objet, ex : «la partie supérieure du bloc moteur». La figure 2.11 présente un échantillon des localisations absolues selon les 3 axes définissant un volume. Bien que toutes les configurations de la figure 2.10 aient une représentation dans la dimension 3, nous ne présentons sur la figure 2.12 que l'équivalent de la figure 2.10.(d) par soucis de clarté.

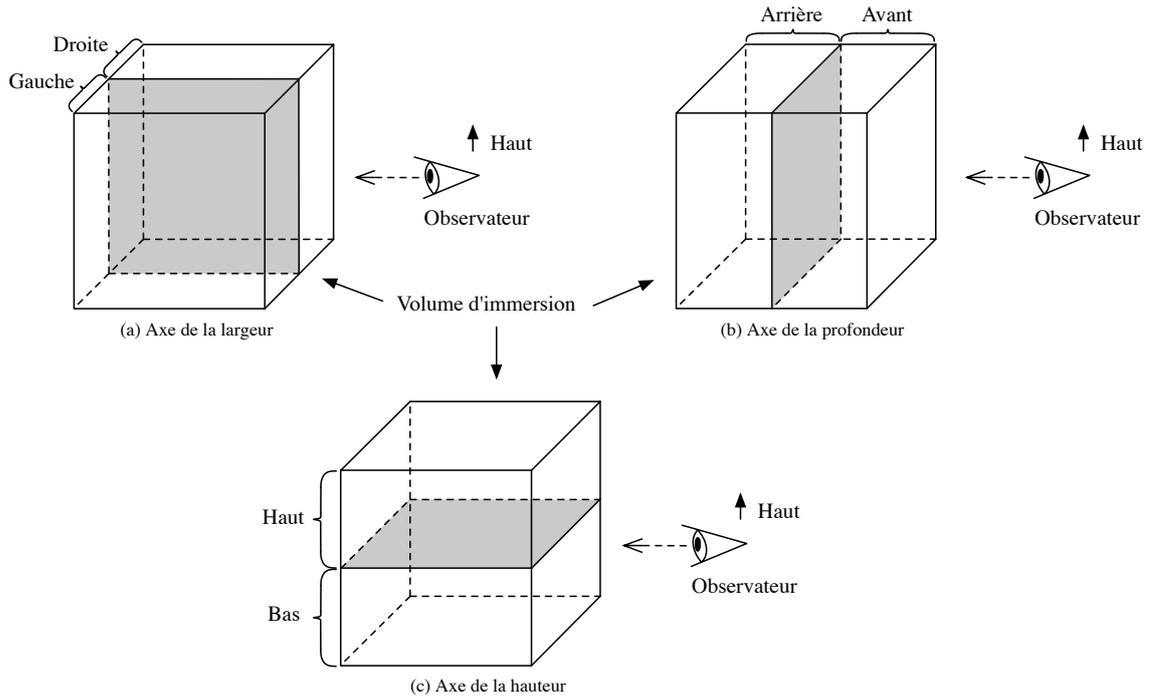


Figure 2.11. Localisations absolues dans un volume selon les 3 axes

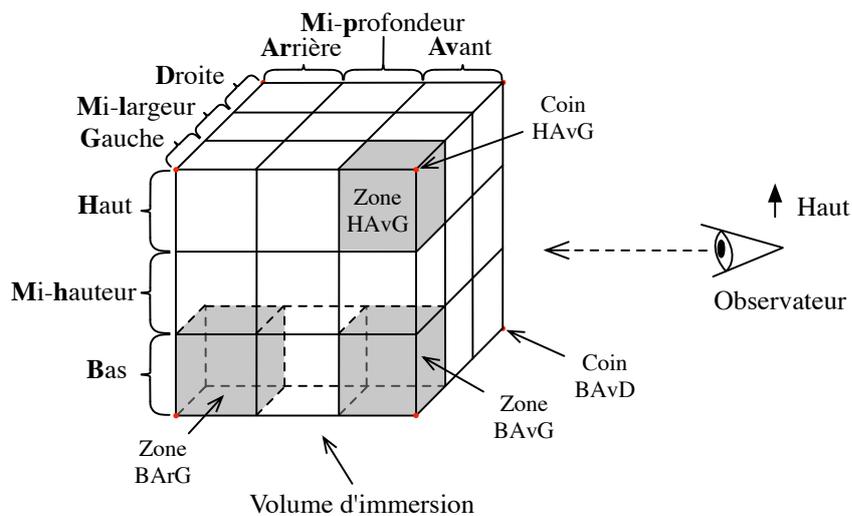


Figure 2.12. Localisations absolues affinées dans un volume

Sur la figure 2.12, le volume d’immersion, ou volume de travail, se subdivise de façon plus précise en exploitant les notions présentées dans la figure 2.11. Comme dans le principe de localisation en dimension 2, la localisation en dimension 3 se présente sous la forme d’un vocabulaire simplifié utilisant les parties notées en gras sur la figure 2.12. Ex : «*Zone HAvG*» correspond à la zone dans la partie **H**aute-**A**vant-**G**auche du volume. La combinaison des termes n’a pas d’importance. Cette notation permet aussi de désigner des éléments connus comme les coins du volume, ex : «*Coin BAuD*» ou son centre «*Point MhMpMl*» que l’on peut aussi désigner par «*Point Central*». Cette terminologie abrégée permet simplement de réduire les descriptions mais elle n’a rien d’obligatoire, l’utilisateur peut toujours utiliser l’expression de localisation complète s’il le désire.

En plus de la localisation, la description doit aussi permettre de donner des spécifications géométriques. Le vocabulaire de cette catégorie est précisé dans la partie suivante.

2.4.2 La géométrie

Dans la description d’objets géométriques tels que les surfaces, les termes de géométrie classique sont indispensables. Ces termes sont par exemple : «*rectangulaire*», «*plat*», «*tangent à*» ...

Cette catégorie de vocabulaire est très riche en sémantique. L’utilisation d’un simple terme ajoute des *propriétés* mathématiques et des contraintes géométriques [Ault, 1999] au modèle. Si la description précise par exemple que la surface est *rectangulaire*, la sémantique du terme *rectangulaire* ajoute indirectement les contraintes géométriques précisant que la surface a quatre côtés, dont les côtés opposés sont parallèles 2 à 2 et que les 2 diagonales sont de même longueur. Mais cela ajoute aussi d’autres propriétés importantes telles que : les côtés opposés sont de même longueur 2 à 2, les diagonales se coupent en leur milieu ou bien encore les côtés consécutifs sont perpendiculaires.

2.4.3 La topologie

La topologie de la surface désirée doit pouvoir être exprimée dans la description. Ainsi une catégorie de vocabulaire spécifique lui est consacré. Les caractéristiques topologiques propres aux surfaces que l’on peut rencontrer sont par exemple : la surface est «*fermée*» ou «*trouée*». Mais ce vocabulaire permet aussi de préciser des lieux topologiques comme l’«*intérieur*» ou l’«*extérieur*» d’une zone de la surface ou de la surface elle-même si celle-ci est fermée.

2.4.4 La morphologie

Le vocabulaire morphologique est destiné à des analogies de formes. Ainsi l’utilisateur peut dans sa description utiliser des termes tels que : «*vallée*» ou «*colline*» ... L’ensemble de ce vocabulaire recueille aussi les termes de formes caractéristiques (ou features) bien

connues et très utilisées en mécanique. Cette catégorie joue donc le rôle de bibliothèque de formes à laquelle l'utilisateur peut se référer lui évitant la description parfois fastidieuse de formes souvent complexes et connues.

Bien que les analogies soient des formes de comparaisons, elles ne sont pas directement exprimées. Les termes qui le permettent sont présentés dans la partie suivante.

2.4.5 Les termes de comparaisons

L'expression directe de comparaisons nécessite un ensemble de termes spécifiques. Ces termes permettent de mettre en relations plusieurs éléments de la surface ou de la scène à travers la comparaison de propriétés communes. Ainsi des termes tels que «*plus ... que*» ou «*autant ... que*» permettent d'exprimer des quantités relatives entre éléments. L'ajout d'une telle catégorie dans le lexique enrichit fortement les possibilités de description tout en rapprochant l'utilisateur d'un ensemble d'expressions plus naturelles.

2.4.6 Les éléments de jonctions

Cette catégorie est essentielle pour une description complète. En effet, elle recueille les termes permettant de lier plusieurs descriptions simples afin d'obtenir des descriptions plus complexes. Ces mots sont typiquement des opérateurs logiques tels que «*et*», «*ou*» ...

2.4.7 Les contraintes

Comment parler d'approche déclarative et de descriptions sans introduire des termes spécifiques à l'expression de contraintes? Bien que l'ensemble du vocabulaire déjà cité induise parfois indirectement la création de nombreuses contraintes, l'utilisateur doit pouvoir aussi directement les exprimer en utilisant un vocabulaire ciblé et immédiatement lié au contexte métier auquel il appartient.

Dans le cadre de la modélisation de surfaces, nous dissocions l'ensemble des contraintes à exprimer en deux catégories distinctes : les contraintes dites «*strictes*», et celles dites «*lâches*».

1. Contraintes «*strictes*» ou «*dures*» :

Les contraintes que nous qualifions de «*strictes*» ou «*dures*» sont les contraintes qui sont généralement nommées *contraintes géométriques* dans la littérature. Ce sont des exigences qui se veulent précises, elles regroupent donc les **contraintes** :

- **de passage** : «*La surface passe par le point P*» (ponctuelle) ou «*La surface passe par la courbe C*» (linéique) ;
- **de non passage** : «*La surface ne passe pas par le point P*» ou «*La surface ne passe pas par la zone Z*» où *Z* est définie comme un sous-volume de la scène ;
- **de tangence** : «*La surface est tangente au plan T*» ;

- **algébriques** [Ault, 1999] : «*La longueur de la surface fait X fois sa largeur*». Ce type de contrainte permet de jouer sur les relations entre paramètres. L'utilisateur peut aussi faire intervenir la notion de conditions par des «*Si ... alors ... sinon ...*» afin d'enrichir ces relations ;
- **de raccordement** : «*La surface se raccorde sur son bord droit avec la surface S1 sans discontinuité*» ;
- **fonctionnelles** [Pace, 1996] : «*la surface doit pouvoir être fixée par quatre vis*». Une telle surface est alors contrainte à admettre quatre trous usinés. Le placement de ces trous dépend de la forme de la surface, mais le choix est généralement fait de manière à maximiser les distances entre ces trous pour une meilleure attache. Cette catégorie de vocabulaire fonctionnel correspond au plus haut niveau conceptuel possible dans une description. elle est indispensable dans une approche déclarative mais reste de très loin la plus difficile à traiter puisque chaque terme utilisé est très riche en sémantique et apporte ainsi beaucoup d'informations au modèle ;
- **d'ingénierie** : «*La surface doit pouvoir être moulée*». Les contraintes d'ingénierie peuvent être de plusieurs ordres et se présentent sous la forme de boîtes noires répondant par un signal booléen (vrai/faux). Elles renferment des calculs difficilement intégrables voire impossibles à intégrer au processus de résolution. Ce type de contraintes définit un espace très vaste. Il va du calcul du coefficient de pénétration dans l'air jusqu'aux contraintes de fabrication en passant par l'estimation financière de la pièce à concevoir.

À travers cet échantillon d'exemples, il est plus facile de percevoir la nécessité de définir des éléments référents (Cf. 2.4.1.1) ou des alias sans lesquels il serait très difficile ou très long de décrire chacune de ces contraintes.

2. Contraintes «lâches»

Les contraintes que nous qualifions de «*lâches*» sont les tendances que veut retrouver l'utilisateur sur la surface après l'ensemble des déformations réalisées pour satisfaire les contraintes plus *strictes*. Ces contraintes acceptent donc une plus grande flexibilité de précision. Les termes de cette catégorie sont : «*bombé*», «*creusé*», «*gonflé*» ...

La notion de contraintes est abordée plus en détails dans la partie 4.3 de ce document. Nous y expliquons comment nous répondons en partie à ce problème en résolvant certaines de ces contraintes par une méthode de déformation contrôlée de la surface ciblée.

2.4.8 La quantification

Alors que tout le vocabulaire déjà cité permet de décrire une surface en partie ou dans sa totalité en précisant des formes, des positions, relatives ou absolues, des tendances de déformations ou des contraintes strictes, il y a un manque cruel de nuances. Ce vocabulaire de quantification est donc essentiel dans une description, il accorde une certaine finition et un certain raffinement aux notions exprimées par l'utilisateur. Les termes de cette catégorie

s'ordonnent selon une échelle qui permet d'estimer et de quantifier. En partant de l'influence minimale vers l'influence maximale voici un échantillon de cet ensemble : «*extrêmement peu*», «*très peu*», «*peu*», «*normalement*», «*assez*», «*très*», «*extrêmement*». La notion de quantification a été traitée à plusieurs reprises à travers l'utilisation d'opérateurs flous [Desmontils, 1995a]. Nous y reviendrons plus en détails dans la partie 3.2.3.4 puis dans la partie 4.3.2 de ce document pour leur exploitation.

2.5 Étude de la description dans un contexte «Mécanique»

2.5.1 Les objectifs

Afin de vérifier s'il était possible d'obtenir des modèles d'objets ou de formes proches de l'original par la simple donnée d'une description écrite en français, nous avons commencé une collaboration inter-équipes dans le cadre du pôle modélisation géométrique du LSIS⁹. Ce projet s'est effectué en collaboration avec l'équipe IMS¹⁰ et l'ENSAM¹¹ d'Aix-en-Provence à travers un projet étudiant de deuxième année qui a donné lieu à un rapport de recherche [Parisot et al., 2002]. Les objectifs fixés se déroulent en trois phases :

1. Décrire de façon *spontanée* des formes dites «*simples*» :



Figure 2.13. Pièce en «U»



Figure 2.14. «Cendrier»



Figure 2.15. «Évier»

Une forme ou un objet est ici qualifié de «simple» s'il n'est construit qu'à partir de formes géométriques connues comme des cylindres ou des parallélépipèdes rectangles (avec des arêtes éventuellement arrondies) ;

2. Trouver une méthode exploitable de description à partir des descriptions spontanées de la phase 1. En déduire :
 - un lexique de tout le vocabulaire classé selon les catégories vues en section 2.4 ;
 - une syntaxe de description de type grammair.
3. Appliquer cette méthode de description à des formes dites «*complexes*» afin de s'assurer de sa robustesse et de son efficacité :

⁹Laboratoire des Sciences de l'Information et des Systèmes

¹⁰Ingénierie, Mécanique, Systèmes

¹¹Ecole Nationale Supérieure d'Arts et Métiers



Figure 2.16. «île»



(a) Vue de face (b) Vue de coté

Figure 2.17. «Capot de Twingo[®]»



(a) Vue de face



(b) Vue de côté



(c) Vue arrière

Figure 2.18. «Rétroviseur de formule RenaultTM»

Une forme ou un objet est ici qualifié de «complexe» s'il contient une partie qui n'est pas une forme géométrique connue comme un cylindre ou un parallélépipède rectangle (avec des arêtes éventuellement arrondies).

Le passage d'une phase à l'autre n'est possible que si la phase courante est achevée. Après une familiarisation des étudiants au concept de la modélisation déclarative, l'étude commence par la première phase du projet : les descriptions spontanées. Le protocole d'expérimentation que nous avons utilisé est expliqué dans la partie suivante.

2.5.2 Protocole d'expérimentation

Le protocole expérimental que nous avons choisi d'appliquer durant cette expérience peut se scinder en quatre étapes :

1. Un premier concepteur modélise des pièces ou *formes modèles* (Fig. 2.13, 2.14, 2.15) qui sont transmises aux étudiants sous la forme d'images et de modèles géométriques ;
2. Les étudiants peuvent virtuellement manipuler les modèles et réalisent une description de chacune des formes proposées ;
3. Chaque description est transmise à un ou deux concepteurs. Ceux-ci ne connaissent pas la forme de départ et n'ont en leur possession que la description fournie par les étudiants. Le ou les concepteurs déduisent de chaque description un ou plusieurs modèles géométriques que l'on nomme aussi *formes résultats*. Plusieurs *formes résultats* sont possibles pour une même description. Les modèles géométriques sont

réalisés par les concepteurs avec l'outil de CAO nommé CATIA^{®12} développé par DASSAULT SYSTEMESTM ;

4. Les *formes résultats* obtenues sont ensuite retournées aux étudiants qui peuvent alors :
 - modifier la description, si le ou les résultats ne sont pas satisfaisants, puis reprendre à l'étape 3 ;
 - valider la description si le ou les résultats sont corrects. Dans ce cas, la description devient *description finale*.

La succession de ces étapes peut se schématiser de la façon suivante faisant apparaître un cycle dans la phase de génération de la description :

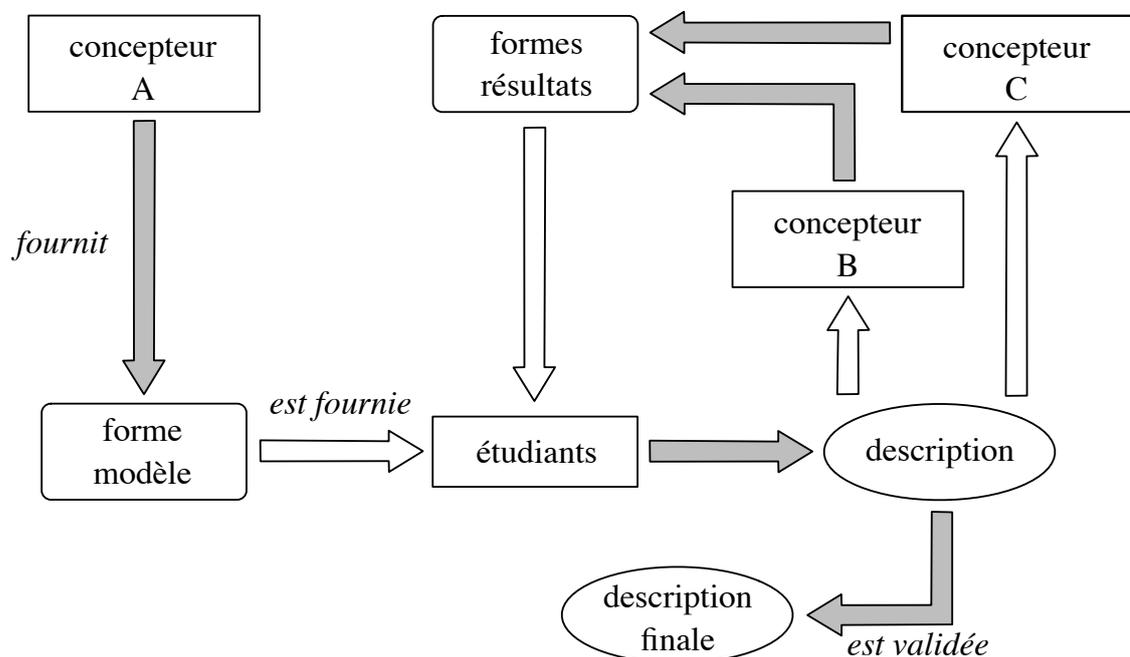


Figure 2.19. Schéma de validation d'une description

Les différents concepteurs de formes impliqués dans l'expérience et les étudiants ont tous des connaissances dans les métiers de la mécanique et de la CFAO. Les descriptions réalisées et interprétées sont donc riches en vocabulaire faisant référence au domaine de la mécanique. C'est donc en appliquant ce procédé de travail que nous avons entrepris toutes les phases de cette expérience.

2.5.3 Première phase

La première phase consiste à trouver une méthode exploitable de description de formes. Pour cela, nous proposons aux étudiants de décrire les formes modèles de façon spontanée [Petit et Morteau, 1996], en faisant des phrases avec le seul souci du résultat de la modélisation.

¹²Conception Assistée Tridimensionnelle Inter Active

Dans ce premier exercice, on aperçoit une certaine tendance des étudiants, eux même concepteurs, à décrire la forme modèle à partir des actions qu'ils seraient menés à réaliser pour la modéliser. Alors qu'il est aussi possible de réaliser une description faisant intervenir une quelconque fonctionnalité de la pièce ou bien encore possible de réaliser une description en tant que simples observateurs. Il apparaît aussi qu'une description spontanée amène les étudiants à sur-contraindre les descriptions afin d'être sûr que le concepteur réussisse à réaliser la forme modèle de façon exacte. Ce qui peut mener à plusieurs cycles dans notre protocole d'expérimentation.

L'ensemble des descriptions finales réalisées pour les trois pièces dites «simples» (Fig. 2.13, 2.14, 2.15) sont en annexe B.2. Voici cependant à titre d'exemple la description de la pièce en «U» (Fig. 2.13) accompagnée de ses modèles résultats :

- *On se place dans un référentiel parallélépipédique défini par les axes x,y,z ;*
- *On se place dans le plan (x,y) , le regard suivant l'axe z décroissant. L'axe des x est horizontal;*
- *La pièce est l'agencement en «U» de trois parallélépipèdes identiques : deux suivant l'axe y et un suivant l'axe x . Chaque parallélépipède est trois fois plus long que large et d'épaisseur une fois et demi sa largeur;*
- *Toutes les arêtes sont arrondies avec le même rayon de raccordement;*
- *On se place, à présent, dans le plan (x,z) , l'axe x étant horizontal et le regard suivant l'axe des y croissants;*
- *On rajoute un cylindre d'axe y au centre du rectangle face à nous, de diamètre égal à la largeur de ce même rectangle et de hauteur égale à trois quarts de cette même largeur;*
- *On perce un trou débouchant au centre du cylindre, de diamètre égal à la moitié du diamètre de ce même cylindre.*

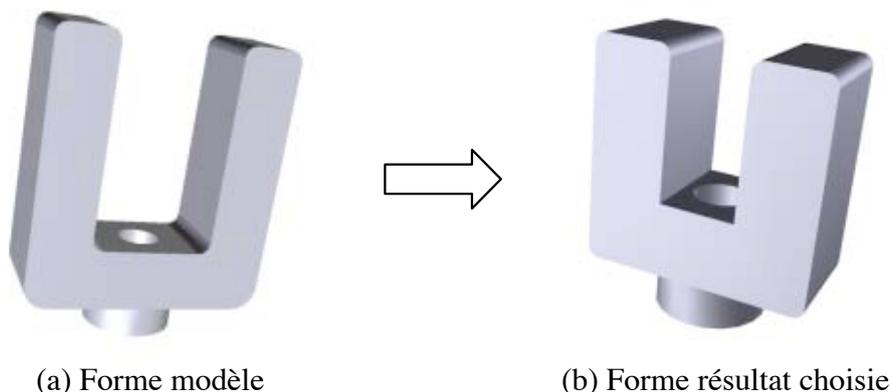


Figure 2.20. Résultats pour la forme : pièce en «U»

Nous pouvons remarquer que certaines parties de la description pourraient être considérées par défaut. C'est le cas, par exemple, de la mise en place du référentiel utilisé et de la vue de

l'observateur. Cependant il s'est avéré, dans certains cas, que l'omission de ces précisions pouvait mener à des ambiguïtés lors de l'interprétation des descriptions. C'est pourquoi nous avons choisi, dans cet exemple, de tout faire figurer dans le texte descriptif.

Les concepteurs ont construit plusieurs pièces répondant, à leur sens, aux exigences faites par les étudiants. Le résultat de la figure 2.20.b est considéré comme correct dans sa plus grande partie face à la forme modèle de la figure 2.20.a. La description est donc validée. Après parfois quelques versions, l'ensemble des descriptions spontanées a été validé et la deuxième phase du projet a pu débuter.

2.5.4 Deuxième phase

Les descriptions obtenues en première phase sont très riches et difficilement exploitables par un système informatique. Dans le but de mettre au point une méthode de description plus exploitable, il a été proposé aux étudiants de réaliser des diagrammes représentant les descriptions réalisées. Le formalisme graphique et structurel que nous leur proposons est l'UML¹³ puisqu'il nous semble bien adapté aux besoins évoqués. Cependant, les étudiants ne connaissaient pas ce formalisme et compte tenu des objectifs et des délais du projet, il n'a pas été possible d'envisager une formation dans ce domaine. Le formalisme UML a donc très vite été abandonné au profit de diagrammes simplifiés.

Ces diagrammes ont pour but de faire ressortir les points essentiels des descriptions tout en éliminant les fioritures du langage courant. Ils ont aussi le net intérêt de faire ressortir du discours les liens entre certains éléments de la description comme par exemple : «*son épaisseur est une fois et demi sa largeur*».

Le diagramme issu de la pièce en «U» est visible sur la figure 2.21, mais il est aussi possible de retrouver tous les diagrammes associés aux formes traitées dans l'annexe B.2.

Les représentations des descriptions des formes *simples* sous forme de diagrammes sont aussi le point de départ de la constitution du lexique. En effet, chaque boîte d'un diagramme isole un terme spécifique de la description. Il a donc été possible de regrouper tous ces termes en catégories dans un lexique comme précisé dans la partie 2.4 de ce chapitre. Le résultat de ce travail est consultable en annexe B.2.2.

Enfin la dernière partie de cette seconde phase s'est focalisée sur l'étude d'une syntaxe descriptive. En effet, à long terme le projet de l'approche déclarative de la modélisation de surfaces est destiné à répondre à des descriptions qui s'orientent vers du langage naturel écrit ou oral. L'approche par diagramme n'est qu'une étape. Notre idée est donc d'avoir une réflexion sur des grammaires de descriptions possibles qui soient beaucoup plus simples que les premières descriptions spontanées, mais aussi plus *rédigées* que des diagrammes. Notre travail se base sur l'étude des grammaires de Chomsky [Chomsky, 1956] et de leur utilisation dans les travaux portant sur le traitement automatique du langage naturel

¹³Unified Modeling Language : formalisme permettant de représenter des architectures objets.

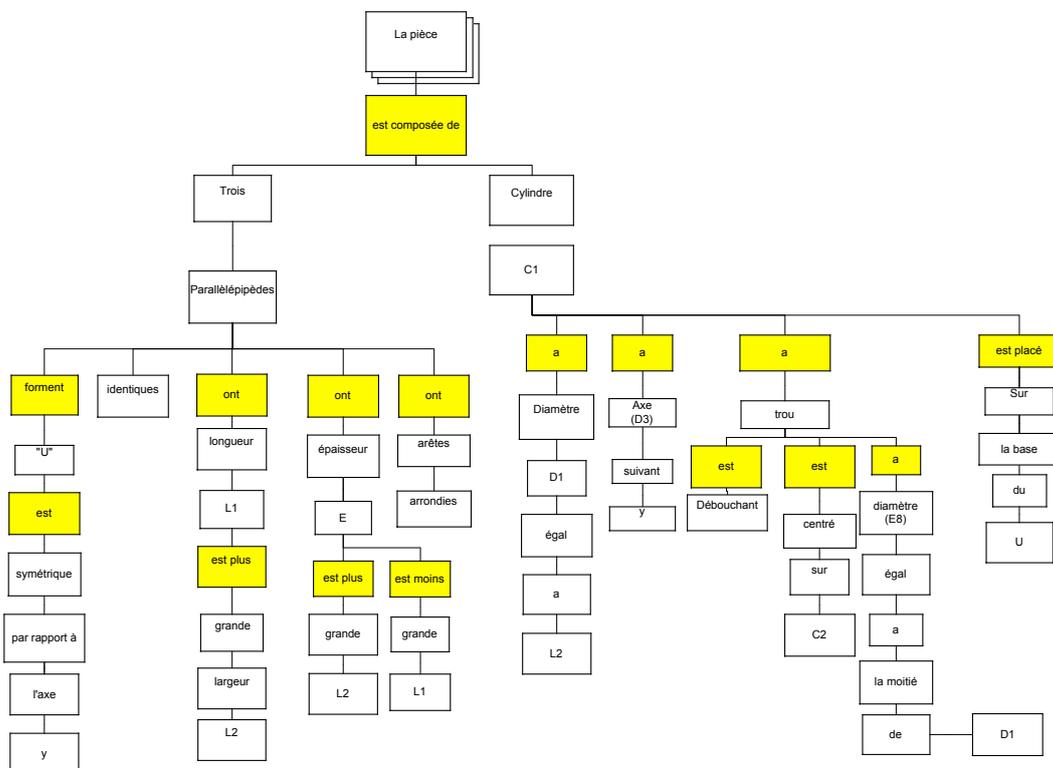


Figure 2.21. Diagramme de description de la pièce en «U»

[Ligozat, 1994]. Après une rapide formation des étudiants à cette technique, les termes ou lexèmes¹⁴ du vocabulaire établi ont été enrichis d'une *nature grammaticale*.

Une première grammaire a alors été mise au point par les étudiants, elle est consultable en annexe B.2.3.1. Les descriptions des trois formes *simples* ont été réécrites en utilisant cette syntaxe grammaticale et elle s'est très vite avérée trop complexe pour des besoins de description. Les phrases étant trop chargées, l'objectif premier de la clarté était perdu.

Une deuxième grammaire a alors été entreprise. L'idée étant d'avoir une description claire n'utilisant que des phrases simples et courtes. Cette grammaire est consultable en annexe B.2.3.2. Pour faire référence à des éléments de la description sans avoir à utiliser de structures du type *subordonnée relative ou compléments circonstanciels* qui alourdissent le discours, le choix a été fait d'introduire une nouvelle catégorie de vocabulaire que nous nommons *identifiants*. Un identifiant est un symbole commençant par une lettre et composé exclusivement de lettres et de chiffres. Il permet de nommer temporairement un ou plusieurs éléments. Ex : *le parallélépipède P1*. *P1* est alors le symbole représentant le parallélépipède en question. Cette notion d'*identifiant* a déjà été abordée dans la partie 2.4.1.1 à travers celle d'*alias*. La nature grammaticale regroupant les verbes a aussi été affinée en fonction

¹⁴Lexème = élément primitif de la grammaire : mot

de leur sémantique. Ainsi on voit apparaître l'ensemble des verbes permettant d'exprimer la comparaison ou bien encore celui des verbes exprimant la localisation.

Toutes les descriptions de formes simples ont été remaniées en utilisant cette syntaxe ainsi que le lexique mis au point. Ce travail a donné trois descriptions plus longues mais exploitables et plus lisibles qui sont consultables en annexe B.2 sous le nom de *description structurée*.

À travers le lexique et cette nouvelle grammaire, il semble possible de réaliser des descriptions exploitables de formes. C'est ce que nous avons voulu vérifier en passant à la troisième et dernière phase de l'expérience.

2.5.5 Troisième phase

Dans cette dernière phase, les travaux se sont penchés sur la description de pièces plus complexes en utilisant les résultats obtenus durant les phases précédentes. Toujours en suivant le même protocole expérimental, mais en se contraignant à n'utiliser que le vocabulaire du lexique et la deuxième syntaxe grammaticale, les étudiants ont entrepris de décrire les trois formes dites *complexes* (Fig. 2.16, 2.17, 2.18). La figure 2.22 présente le résultat final obtenu, toujours avec CATIA® et après quelques cycles de description, sur la forme «capot de Twingo®». Tous les autres résultats sont présentés en annexe B.2.

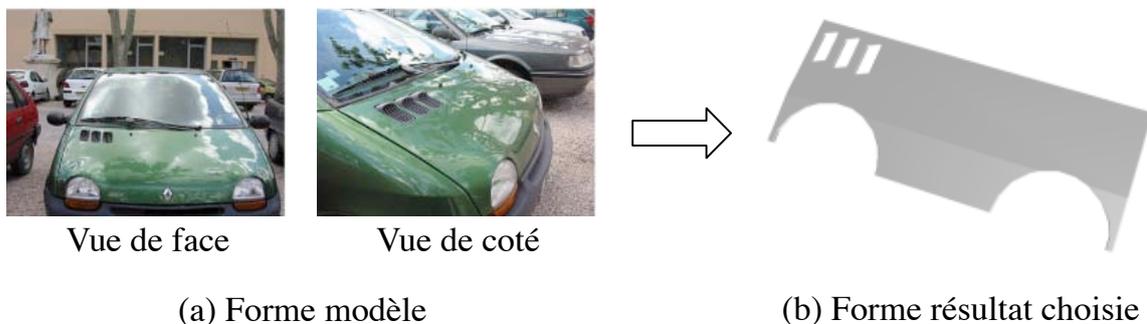


Figure 2.22. Résultats pour la forme : capot de Twingo®

À l'issue de cette troisième phase, il s'avère que les descriptions sont très claires et qu'elles permettent d'obtenir en peu de cycles de description des résultats très proches de ceux souhaités. De part leur structure n'utilisant que des phrases courtes, les descriptions sont très allongées du fait des nécessaires répétitions. Cette constatation fait naître une perspective de recherche portant sur le module d'interfaçage devant supporter ce pseudo-langage. L'option d'une description interactive paraît être une piste de travail intéressante.

2.6 Conclusion

Ce chapitre a présenté les difficultés et les besoins effectifs de la modélisation déclarative de surfaces en terme de description. Il s'avère que l'aspect multimodal de la description permet de rapprocher l'utilisateur de la machine en lui proposant des moyens d'expression plus adaptés à ces besoins. Néanmoins, particulièrement attachée au contexte de travail et à la subjectivité de l'utilisateur, la description se révèle être un exercice difficile. Il est cependant possible, en introduisant la notion de domaine d'utilisation, ou contexte métier, d'obtenir des descriptions plus adaptées et utilisant un vocabulaire moins ambigu et plus proche des besoins de l'utilisateur. Ainsi, l'orientation de nos travaux se porte sur le contexte métier de la mécanique pour lequel nous proposons un pseudo-langage de description s'appuyant sur la langue française. Ce pseudo-langage semble être suffisant pour répondre aux besoins de la modélisation déclarative de surfaces, même si celui-ci ne peut être considéré comme achevé.

Une description est avant tout riche en sémantique. C'est pourquoi, après avoir évoqué dans ce chapitre les besoins descriptifs de la modélisation déclarative de surfaces, nous allons parler dans le chapitre suivant de l'organisation et de la représentation des connaissances de cette description dans le but de leur exploitation.

MODÈLES SÉMANTIQUES

3.1 Introduction

L'approche déclarative de la modélisation de surfaces doit permettre de générer une ou plusieurs surfaces répondant à la description fournie par un utilisateur. À travers cette description, le concepteur exprime beaucoup d'informations qu'il est nécessaire de conserver dans la ou les solutions. Ces informations doivent alors être extraites de la description puis structurées de manière à être exploitables par tous les traitements suivants de notre approche déclarative mais aussi par tous ceux susceptibles d'être plus tard appliqués aux solutions générées. Dans une première partie, nous introduisons un premier modèle dit *sémantique* obtenu après un traitement de *traduction* appliquée sur la description fournie par l'utilisateur. Cette traduction fait intervenir la notion de *modules d'interface* donnant une certaine souplesse aux formats de description permettant ainsi une approche multimodale du problème.

Avant d'aborder les traitements, il est important de bien connaître les structures de données sur lesquelles nous travaillons. C'est pourquoi la partie suivante détaille l'architecture commune des données que nous utilisons pour structurer la connaissance de l'ensemble des modèles *sémantiques*. Le concept de *zones*, autour duquel gravitent tous nos travaux, y est abordé. Nous précisons ensuite ce que l'on appelle base de connaissances, ou base déclarative, en détaillant sont contenu.

Ce n'est qu'après que l'on se penche véritablement sur l'ensemble des traitements de ces données en commençant par la *complétion sémantique*. C'est dans cette partie qu'est expliqué pourquoi et comment, à partir d'un modèle sémantique, nous obtenons plusieurs modèles sémantiques *généralisés* à travers deux processus successifs : la complétion *symbolique* et la complétion *numérique*. Nous poursuivons, avant de conclure, par la présentation de notre moteur de *planification* qui permet d'obtenir un ensemble de modèles *patrons* indispensables pour la construction des modèles géométriques solutions. Chacun de ces deux processus est illustré par un exemple simple afin de mieux percevoir leur déroulement.

3.2 De la description vers la sémantique

À ce stade de l'analyse, une description a été fournie au système. L'interprétation de cette description dans les traitements suivants passe par une traduction des idées et des concepts exprimés vers un langage sémantique structuré. Pour cela et dans le but de garder un système très évolutif avec une connotation multimodale forte, nous avons fait émerger la notion de *modules d'interface* présentée dans cette première partie.

3.2.1 Le moteur de traduction sémantique

Le moteur de traduction sémantique peut être vu comme une boîte noire à plusieurs entrées et une sortie (Fig. 3.1). Il prend en entrée un ou plusieurs *signaux descriptifs* et il génère en sortie un modèle dit *modèle sémantique*.

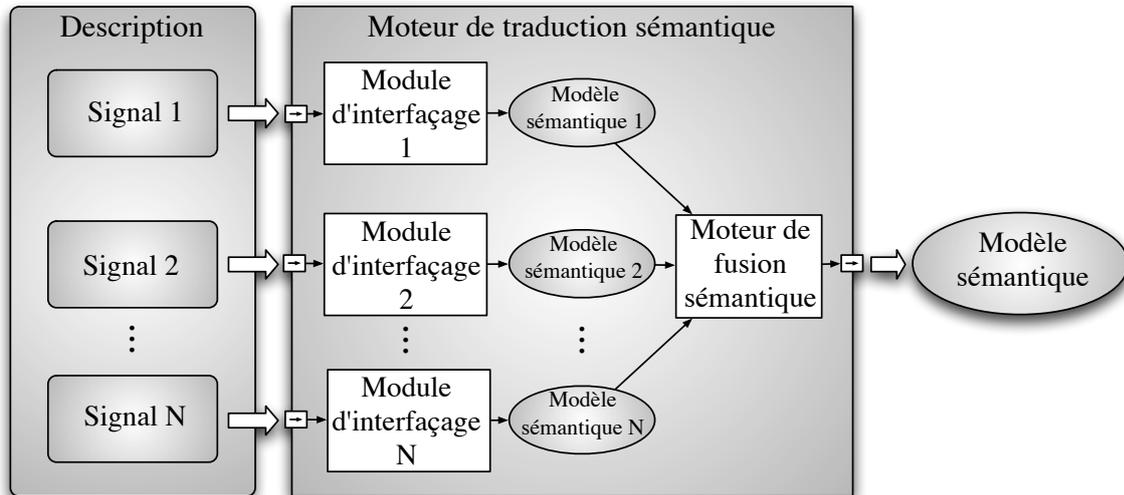


Figure 3.1. Moteur de traduction sémantique et ses modules d'interface

Un *signal descriptif* peut se définir comme un flux d'informations émis par un utilisateur. Ce flux code des données de nature descriptive et peut se présenter sous plusieurs formes : un son dans le cas d'une description orale, un texte dans le cas d'une description écrite ou des coordonnées spatiales provenant de capteurs optiques dans le cas d'une cage de réalité augmentée avec capteurs de mouvement par exemple.

Le moteur de traduction sémantique est prévu pour gérer plusieurs signaux descriptifs en même temps. En effet, chaque entrée du moteur est directement attachée à un type de signal donné qu'il a la capacité de traiter par l'intermédiaire d'un *module d'interface* spécifique. Un *module d'interface* est donc un système de traduction qui prend en entrée un signal descriptif donné et qui génère en sortie le modèle sémantique associé dont la structure est détaillée dans la partie 3.2.2. Cette extraction et cette structuration de la sémantique dépendent du type de signal en entrée. Ainsi un module d'interface destiné à traiter

des textes écrits en français sera basé sur des techniques de traitement automatique de la langue écrite française en utilisant des méthodes basées sur les grammaires [Ligozat, 1994] ou bien des méthodes plutôt statistiques [Spriet and El-Beze, 1998]. Et celui devant traduire des descriptions orales en français utilisera des techniques de reconnaissance de la parole [De Mori, 1998].

Chaque module d'interfaçage génère donc un modèle sémantique. Le *moteur de fusion sémantique* permet de rassembler toutes les informations en un seul modèle sémantique afin de centraliser les données et de supprimer les redondances possibles.

L'ajout d'entrées dans le moteur de traduction sémantique amplifie son aspect multimodal et le rend évolutif. À travers cette architecture, chaque nouveau module d'interfaçage peut être considéré comme un plugiciel¹ qui a l'avantage d'enrichir le moteur de traduction sémantique sans avoir à en modifier le code. Il est ainsi possible d'activer ou de désactiver certains modules, d'en privilégier certains au profit d'autres et donc d'avoir une grande flexibilité dans la configuration des traductions possibles des descriptions à traiter.

Le passage de la description à sa première représentation sémantique qu'est le modèle sémantique se déroule donc en deux étapes :

1. la *traduction sémantique* spécifique à un signal descriptif qui est réalisé par un module d'interfaçage donné. Cette traduction faite sur toutes les N entrées actives du moteur de traduction sémantique génère alors N modèles sémantiques ;
2. la *fusion sémantique* qui est à la charge du moteur de fusion sémantique consiste à rassembler tous les modèles sémantiques obtenus pour n'en faire qu'un.

Durant ces deux étapes de traduction et de fusion des données, il est déjà possible de détecter des contradictions, des incohérences ou des ambiguïtés de premier ordre comme des références à des objets non déclarés ou des identifiants identiques pour des objets différents. L'utilisateur est alors invité à reprendre la ou les descriptions qui est ou qui sont à la source du désordre. Ce premier niveau de vérification garantit la génération d'un modèle sémantique ayant une base cohérente au sens de la traduction.

S'il n'y a qu'un seul signal descriptif en entrée du moteur de traduction, ou qu'un seul module d'interfaçage est activé, la fusion sémantique ne consiste qu'à réémettre l'unique modèle sémantique obtenu.

Dans le cadre de cette thèse, l'objectif fixé du passage d'une description ou plusieurs descriptions à un modèle sémantique reste à un niveau d'analyse conceptuel. Les besoins ayant été ainsi dégagés et l'architecture établie, une collaboration a été amorcée autour du pôle modélisation géométrique du LSIS avec l'équipe INCA². La piste de travail évoquée a été l'utilisation de méthodes de configuration [Grandcolas and al., 2003] et un travail de DEA portant sur l'«*analyse syntaxico-sémantique de textes descriptifs au moyen de la configuration*» [Estratat, 2003] a été réalisé. Les premiers résultats étant encourageants, ils

¹ou plug-in en anglais

²INférence, COntraintes et Applications

sont actuellement poursuivis au travers d'une thèse intitulée «*Configuration, Isomorphismes structurels et Application aux textes descriptifs*» [Estratat et Henocque, 2005]. À terme, ces travaux ont pour but la réalisation d'un premier module d'interfaçage permettant de générer un modèle sémantique simplifié à partir d'une description écrite dans un pseudo-langage de description basé sur la langue française similaire à celui mis au point dans la partie 2.5. L'organisation des connaissances du modèle sémantique et de ses généralisations respecte une structure précise et adaptée à l'approche déclarative de la modélisation de surfaces. Cette structure est présentée dans la partie suivante.

3.2.2 Structure du modèle sémantique et de ses généralisations

Le modèle sémantique, obtenu lors de la traduction sémantique de la description, évolue en modèles sémantiques intermédiaires puis en modèles sémantiques généralisés (Fig. 3.2). Cette évolution consiste à compléter le modèle sémantique issu du procédé de traduction à partir des données stockées dans la base de connaissances que l'on présente dans la partie suivante. Si le modèle sémantique n'a pas besoin d'être complété, il est considéré comme étant déjà généralisé. Dans le cas contraire, l'ajout des données se fait par l'application d'un ensemble de règles. S'il existe dans la base de connaissances, plusieurs configurations ou plusieurs symboles possibles pour une même donnée, il est créé autant de modèles sémantiques intermédiaires que de possibilités. Ce processus itératif ainsi que les règles qui s'y rapportent sont détaillés dans la partie 3.2.4. Nous appelons ce traitement la *complétion sémantique*.

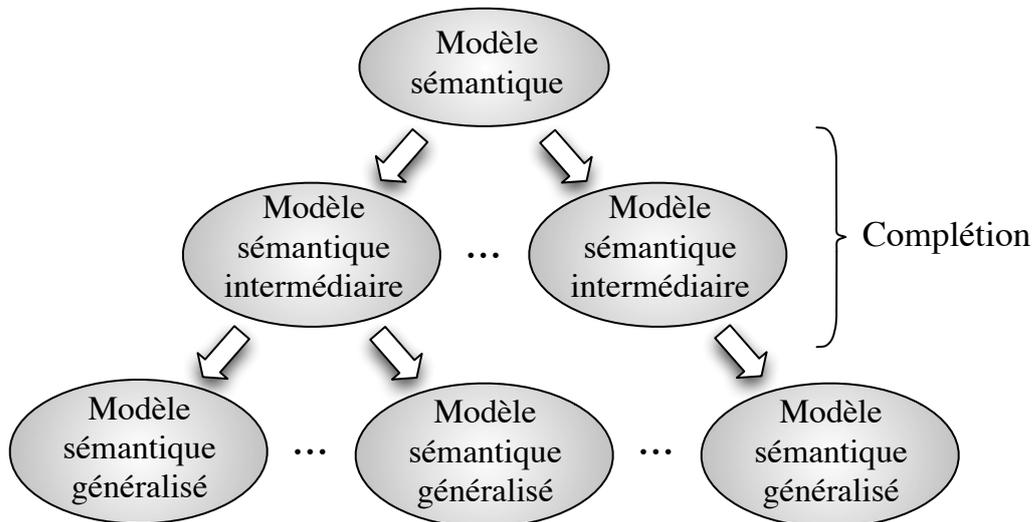


Figure 3.2. Arbre de déclinaison des modèles sémantiques ou arbre des solutions

Cet arbre de déclinaison des modèles sémantiques est le début de ce que l'on nomme *arbre des solutions* et qui constitue à la fin de tout le processus déclaratif que nous proposons, l'espace des solutions.

L'organisation des données des trois modèles sémantiques dont nous venons de parler, est la même durant toute l'évolution des modèles. Dans cette partie, nous détaillons cette structure qui est donc aussi celle de la version la plus évoluée des modèles sémantiques : le *modèle sémantique généralisé*. Le ou les modèles sémantiques généralisés constituent la connaissance même de la ou des formes finales obtenues. De ce fait ils doivent contenir à la fois les informations de *haut niveau* issues de la description de l'utilisateur, mais aussi les informations dites de *bas niveau* déduites du moteur de complétion sémantique.

C'est le langage XML³ que nous choisissons pour exprimer la sémantique traduite de la description. Par ses possibilités et son efficacité dans le domaine de la structure des connaissances, XML est devenu un standard et s'utilise de plus en plus pour le transfert des données. Son utilisation est alors un atout majeur pour l'évolutivité et la compatibilité des modèles. La structuration XML d'un modèle sémantique généralisé que nous avons mise au point répond à une organisation précise. La notion de définition de document type, ou DTD⁴, qui accompagne XML sert à formaliser cette structure. La DTD joue alors le rôle d'une grammaire syntaxique permettant de vérifier la conformité du document XML utilisé. Celle dont nous nous servons actuellement dans notre démarche est consultable en annexe C.1. Elle structure la connaissance représentée sur le schéma de la figure 3.3 qui utilise le formalisme UML.

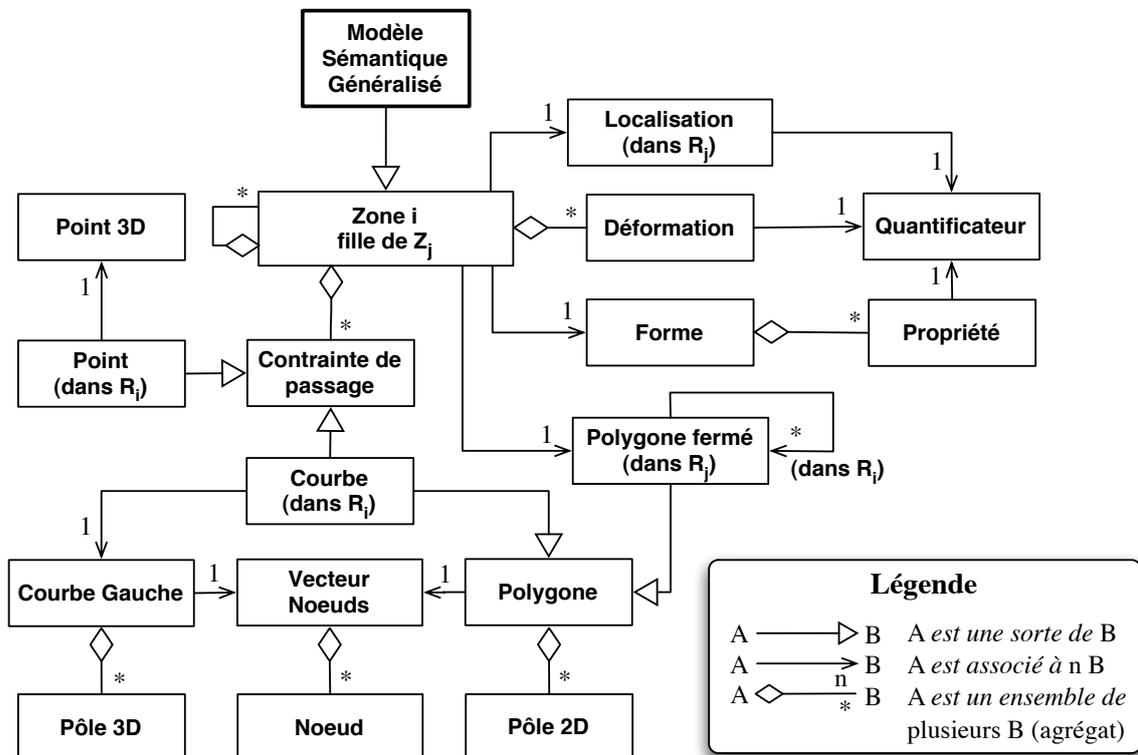


Figure 3.3. Schéma UML de la structure d'un modèle sémantique généralisé

³eXtended Markup Language

⁴Document Type Definition

L'organisation générale des connaissances que nous proposons gravite autour du concept de «zones». Nous définissons une zone de la surface par une courbe NURBS fermée. Le polygone de contrôle de cette courbe est exprimé dans le plan paramétrique de la surface, c'est pourquoi, nous parlerons de *zone paramétrique*. Ce choix a été réalisé car la description est entièrement donnée en entrée du processus. Elle donne et localise des formes sur la surface sans tenir compte du procédé de construction et des déformations appliquées au polyèdre de contrôle. Ainsi, la description se réfère à un invariant attaché à la surface. Le plan paramétrique remplit parfaitement cette tâche.

Chaque zone peut elle-même être constituée d'autres zones plus petites qu'elle contient entièrement. Une surface peut donc se décrire sur plusieurs niveaux faisant apparaître la notion de zones filles et de zones mères dont la structure finale est un arbre où chaque nœud est une zone. Cet arbre est nommé *arbre structurant* et sera noté \mathcal{A}_s dans toute la suite de notre étude. La figure 3.4 introduit cette notion sur un exemple où la surface est constituée d'un total de sept zones (la zone racine est comptée). Parmi ces zones, Z2 est fille de Z1, les zones Z4 et Z5 sont filles de Z3 et les zones Z1, Z3 et Z6 sont filles de Z0 qui est la zone racine.

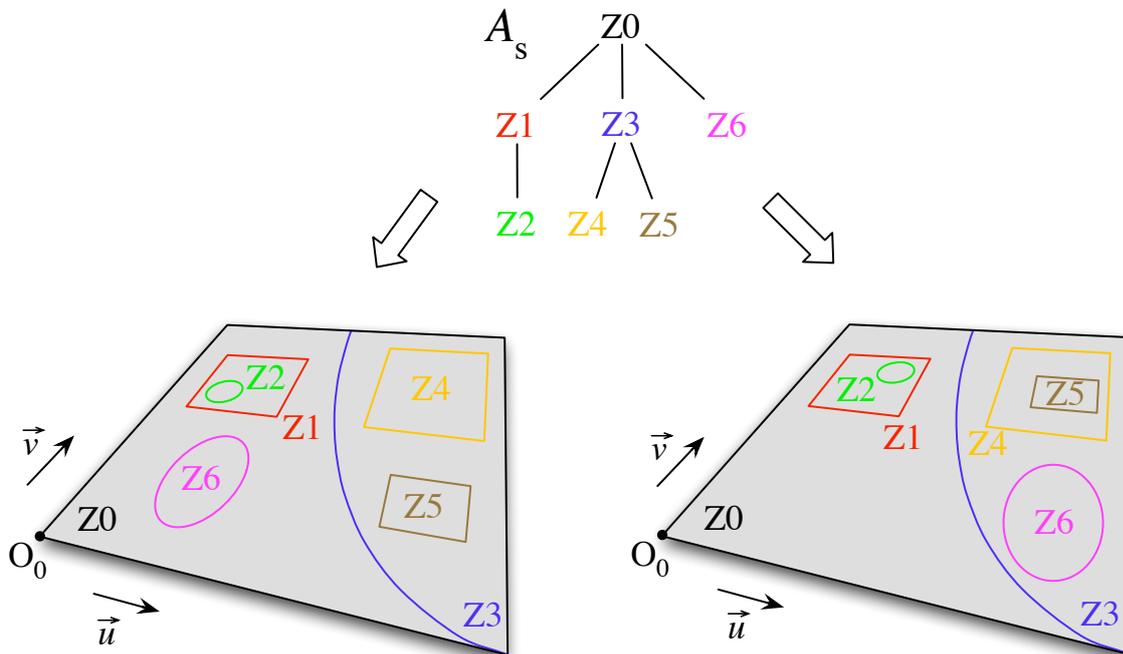


Figure 3.4. Arbre structurant identique pour des localisations différentes de zone

De plus, cet exemple met en avant le fait que l' \mathcal{A}_s est totalement indépendant de la localisation des zones. Il ne représente que la structure qui lie les zones entre elles, ainsi il est possible que deux zones se chevauchent sans pour autant que l'une soit fille de l'autre. L' \mathcal{A}_s se définit donc directement de la structure XML du modèle sémantique.

Chaque zone de cet arbre *structurant* engendre localement un nouvel espace paramétrique, défini dans celui de sa zone mère, par la boîte englobante de son polygone de contrôle (Fig. 3.5). Nous notons R_i l'espace paramétrique local engendré par la zone Z_i . Chacun de ces plans paramétriques est attaché à un repère orthonormal local $(O_i, \vec{u}_i, \vec{v}_i)$ borné tel que $u_i \in [0, 100]$ et $v_i \in [0, 100]$, O_i étant le sommet bas-gauche de la boîte englobante considéré comme origine du repère local. Ce choix de bornes a été fait pour directement exprimer la localisation sous la forme de pourcentage. Cette échelle est d'ailleurs utilisée pour l'ensemble des quantifications de notre processus déclaratif.

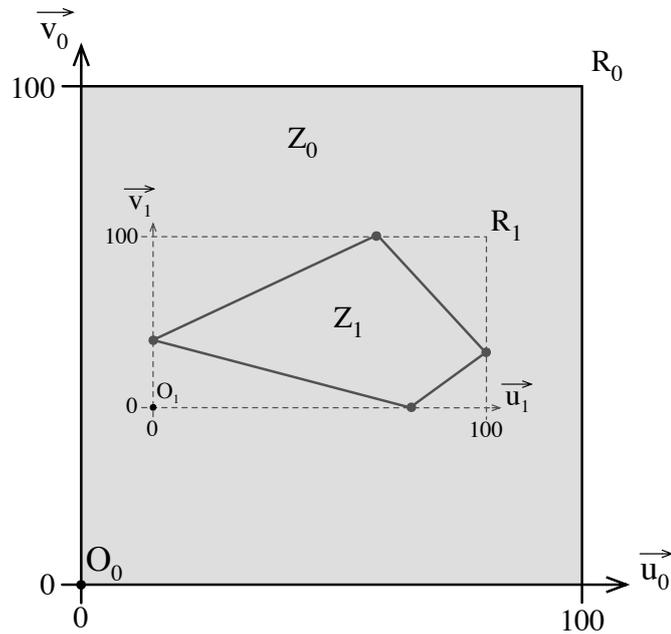


Figure 3.5. Plan paramétrique \mathcal{R}_0 associé à \mathcal{S}

Le cas de la zone racine Z_0 est un peu différent en terme de localisation. En effet, O_0 est connue comme le couple $(0, 0)$ et le plan de travail \mathcal{R}_0 de Z_0 n'est autre que le plan paramétrique sur lequel les futurs modèles géométriques solutions seront définis. Ainsi $u_0 \in [0, 100]$ et $v_0 \in [0, 100]$ correspondent aux intervalles paramétriques d'un carreau de surface NURBS. La figure 3.5 illustre le plan de travail \mathcal{R}_0 dans lequel est définie une zone Z_1 de forme quelconque accompagnée de son référentiel local \mathcal{R}_1 .

Notons que chaque zone n'ayant pas forcément une forme rectangulaire dont les côtés sont parallèles à ceux du plan paramétrique, il est très souvent possible d'obtenir des repères \mathcal{R}_i dont l'origine O_i n'appartient pas à la zone Z_i , ce qui est par exemple le cas de la figure 3.5. Néanmoins, nous considérons que c'est l'intérieur de la forme de Z_i qui est l'espace valide de travail. Cette situation est localement équivalente à une restriction de domaine paramétrique.

Une surface dite *déclarative* se définit à travers un modèle sémantique généralisé comme la *zone* racine notée \mathcal{Z}_0 de cet arbre de données dont le repère local est R_0 . Formellement, une **Zone** \mathcal{Z}_i , fille de \mathcal{Z}_j , se définit par le sextuplé $\{\mathcal{Z}, \mathcal{F}, L, \mathcal{P}, \mathcal{D}, \mathcal{C}\}$ tel que :

- \mathcal{Z} : Ensemble de toutes les zones filles de \mathcal{Z}_i . \mathcal{Z} peut être vide, dans ce cas, \mathcal{Z}_i n'a pas de zone fille. Une zone fille est dite valide si sa forme est entièrement incluse dans celle de sa zone mère. Dans le modèle XML, une zone fille se définit par les balises `<Zone> ... </Zone>` et doit être constituée de l'ensemble des éléments $\mathcal{Z}, \mathcal{F}, L, \mathcal{P}, \mathcal{D}$ et \mathcal{C} . C'est l'imbrication des ensembles \mathcal{Z} , c'est-à-dire des balises `<Zone>` qui induit l'arbre structurant \mathcal{A}_s ;
- \mathcal{F} : Information symbolique définissant la forme de \mathcal{Z}_i . Elle se définit par les balises `<Forme> ... </Forme>` dans l'architecture XML. Les identifiants possibles sont ceux prédéfinis comme «CARREE», «RONDE» ..., dans la base de connaissances (Cf. 3.2.3.3). Cette information est utilisée pour la construction du polygone de contrôle de \mathcal{Z}_i lors du processus de complétion numérique ;
- L : Information symbolique définissant la localisation de \mathcal{Z}_i dans \mathcal{R}_j . Dans la structure XML, elle se définit par les balises `<Localisation> ... </Localisation>`. Ces dernières peuvent elles-mêmes contenir les balises `<Absolue> ... </Absolue>` ou `<Relative> ... </Relative>`. Les identifiants possibles sont ceux prédéfinis dans la base de connaissances (Cf. 3.2.3.2) tels que «PARTIE DROITE» ou «ZONE HG» pour les localisations absolues et «A GAUCHE DE» ou «AU DESSUS DE» pour les localisations relatives. Cette information est utilisée pour la construction du polygone de contrôle de \mathcal{Z}_i lors du processus de complétion numérique ;
- \mathcal{P} : Polygone ou polygone de contrôle fermé défini dans \mathcal{R}_j . Dans le cas où la forme est complètement anguleuse comme un triangle, un carré, un rectangle ou plus généralement un polygone, \mathcal{P} est considéré comme un polygone fermé défini seulement par des sommets, les frontières de \mathcal{Z}_i dans \mathcal{R}_j sont donc directement \mathcal{P} . Dans le cas où la zone présente au moins une portion courbe, alors \mathcal{P} est considéré comme le polygone de contrôle d'une courbe NURBS plane fermée. C'est alors cette dernière qui permet de délimiter les frontières de \mathcal{Z}_i dans \mathcal{R}_j . Dans la suite de ce document nous ne parlerons, par abus de langage, que de polygone de contrôle d'une zone en parlant de \mathcal{P} afin de rester le plus général possible. \mathcal{P} est éventuellement aussi associé à d'autres polygones ou polygones de contrôle fermés qui sont exprimés sur \mathcal{R}_i , à l'intérieur des frontières de la courbe qu'il définit. Ces autres frontières délimitent des trous dans la forme de \mathcal{Z}_i permettant ainsi d'engendrer des zones dont la forme est topologiquement contrôlable. La figure 3.6 montre un exemple de zone \mathcal{Z}_1 trouée et fille de \mathcal{Z}_0 qu'il est ainsi possible d'obtenir :

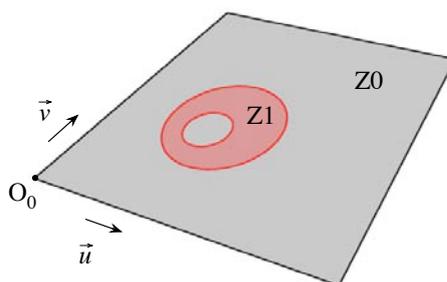


Figure 3.6. Exemple de zone à forme trouée : Z_1

Cette notion de frontière de zone exprimée par \mathcal{P} est très importante dans notre approche car c'est elle qui est directement utilisée dans l'ensemble des calculs réalisés dans les processus de *planification* et de *résolution/construction*. Ainsi, si \mathcal{P} n'est pas explicitement donné dans la description, il est construit par le moteur de complétion numérique à partir des informations de forme (\mathcal{Z}) et de localisation (L) fournies par l'utilisateur. Ce sont les balises `<Polygone> ... </Polygone>` qui expriment \mathcal{P} dans la structure XML. Ces balises prennent pour attribut le degré de la frontière définissant la forme. S'il n'est pas indiqué, il est fixé par défaut à 1. Dans le cas où le degré est égal à 1, \mathcal{P} est considéré comme un polygone, s'il est par contre strictement supérieur à 1, il est considéré comme un polygone de contrôle et un vecteur de nœuds doit aussi être indiqué. Les balises XML définissant le polygone peuvent elles-mêmes contenir plusieurs balises `<Moins> ... </Moins>` qui définissent de la même manière les trous dans la zone ;

- \mathcal{D} : Ensemble des déformations attachées à Z_i (contraintes lâches). Ces déformations sont par exemple *bombée* ou *creusée*. Elles sont souvent accompagnées d'un quantifiant qui permet d'en contrôler l'amplitude. Dans l'architecture XML, les balises `<Deformation> ... </Deformation>` permettent d'attacher des déformations à une zone. Les identifiants possibles, comme «BOMBE» ou «COURBE», sont prédéfinis dans la base de connaissances par l'intermédiaire du *graphe de dépendances fonctionnelles*⁵ (Cf. 3.2.3.1) ;
- \mathcal{C} : Ensemble des contraintes dites *dures* attachées à Z_i . Ces contraintes doivent pouvoir, à terme, être de tout ordre comme cela a été décrit dans le chapitre précédent. Dans le cadre de cette thèse, nous nous limitons au passage par un ou plusieurs points de l'espace. Pour cela, les informations nécessaires sont les coordonnées du ou des points de l'espace à atteindre. Il est aussi possible d'indiquer les paramètres exacts en (u, v) qui doivent être utilisés pour le point de passage ainsi que la précision minimale que doit respecter chaque contrainte. La notion peut être étendue au passage par une courbe gauche en discrétisant la courbe destination selon une certaine

⁵Graphe définissant les liens entre déformations dont chaque nœud est un identifiant de déformation disponible dans le système déclaratif

précision [Pernot, 2004]. La contrainte linéique devient donc une suite de contraintes ponctuelles. Dans ce cas, le système doit connaître la courbe gauche à atteindre. Pour cela, le modèle stocke ses pôles, son degré et son vecteur de nœuds. Il est aussi possible de définir la courbe des paramètres que doit suivre la contrainte linéique directement dans le repère local de la zone du plan paramétrique. Dans la structure XML, ce sont les balises `<Contraintes> ... <\Contraintes>` qui permettent de donner au système les contraintes. Elles peuvent elles-mêmes contenir les balises `<Point> ... <\Point>` ou `<Courbe> ... <\Courbe>`.

La syntaxe XML exacte de la définition d'un modèle sémantique et plus généralement d'un modèle sémantique généralisé, est consultable via la DTD dans l'annexe C.1. Voici un exemple simple de modèle sémantique généralisé avec son résultat en figure 3.7 :

<pre> <?xml version="1.0"?> <!DOCTYPE Surface SYSTEM "ModeleSemantique.dtd"> <Surface Id="Z0"> <Localisation> <Absolue Id="TOTALE"/> </Localisation> <Forme Id="CARREE"/> <Deformation Id="NONE"/> <Polygone> <Sommet Id="S0" U="0" V="0"/> <Sommet Id="S1" U="0" V="100"/> <Sommet Id="S2" U="100" V="100"/> <Sommet Id="S3" U="100" V="0"/> </Polygone> <Zone Id="Z1"> </pre>	Suite ->	<pre> <Localisation> <Absolue Id="PARTIE DROITE"/> </Localisation> <Forme Id="TOTALE"/> <Deformation Id="BOMBE"> <Quantificateur Id="ASSEZ" Ecart="4.67" Valeur="66.0000"/> </Deformation> <Polygone> <Sommet Id="S0" U="50" V="0"/> <Sommet Id="S1" U="50" V="100"/> <Sommet Id="S2" U="100" V="100"/> <Sommet Id="S3" U="100" V="0"/> </Polygone> </Zone> </Surface> </pre>
---	----------	---

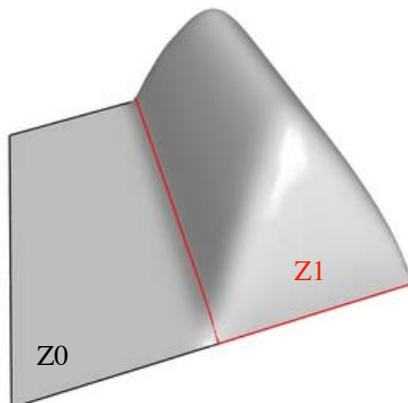


Figure 3.7. Résultat correspondant à l'exemple de modèle sémantique généralisé

Toute *propriété*, *localisation* ou *déformation* (Fig. 3.3) peut être contrôlée par un quantificateur. Celui-ci a pour rôle d'amplifier ou au contraire d'atténuer un phénomène. Pour cela, l'utilisateur emploie généralement dans sa description des adverbes comme «*peu*», «*faiblement*» ou «*fortement*» qu'il peut appuyer avec l'adverbe «*très*» et ainsi exprimer qu'une déformation est «*très faiblement*», «*très très peu*» ou au contraire «*très fortement*» exprimée. Les balises <Quantificateur> ... </Quantificateur> permettent d'exprimer la quantification dans la structure XML. Les identifiants qui peuvent être utilisés sont définis dans la base de connaissances (Cf. 3.2.3.4). Afin de prendre en compte l'insistance possible d'une quantification, nous faisons apparaître un poids dans la structure XML des modèles sémantiques. Celui-ci est directement lié au nombre de termes «*très*» se rapportant au quantificateur que l'utilisateur veut appuyer.

3.2.3 La base de connaissances

La base de connaissances, ou base déclarative, renferme l'ensemble des données pouvant intervenir durant la complétion sémantique et la planification. Elle est exprimée en XML et peut être consultée en annexe C.2 où l'on trouve aussi la DTD qu'elle respecte.

3.2.3.1 Le graphe de dépendances fonctionnelles et déformations disponibles

La base de connaissances contient la définition du *graphe de dépendances fonctionnelles*, noté \mathcal{G}_{df} . Les nœuds de ce graphe sont les déformations disponibles dans le système et les liens qui les relient permettent de fixer l'ordre dans lequel elles doivent être réalisées. Soit A et B deux nœuds de ce graphe, les liens entre A et B peuvent être de deux types :
→ Dépendance simple : notée comme une simple flèche (ou lien orienté) allant de A vers B . Elle signifie que A doit être appliquée avant B . Une dépendance simple reflète une contrainte d'ordonnement d'application des déformations. Ainsi, un lien simple d'une déformation sur elle-même n'a aucun sens puisque un tel lien ne contient pas l'information d'orientation qui traduit la contrainte d'ordonnement ;

$$A \longrightarrow B$$

↔ Double dépendance : notée comme une flèche double (ou lien double). Elle est équivalente à avoir deux dépendances simples, l'une allant de A vers B et l'autre allant de B vers A ce qui est finalement équivalent à dire qu'il n'y a pas de contrainte d'ordre entre A et B . Ce lien double signifie cependant qu'il existe une influence réciproque entre A et B . C'est-à-dire que A appliquée avant B donne un résultat différent de l'application de B avant A . Il reflète donc la non commutativité entre deux déformations.

$$A \longleftrightarrow B$$

Ce double lien peut aussi lier une déformation à elle-même. Prenons l'exemple de la déformation «Bomber», deux déformations «Bomber» peuvent donner des résultats différents si on les applique sur une même partie d'une surface dans des ordres différents. Cela correspond à deux déformations identiques sur deux zones ayant un recouvrement. Cet exemple est d'ailleurs illustré un peu plus loin avec la figure 3.16.

Remarque 3.2.1 – Une déformation qui n’a pas de lien double la reliant à elle-même est considérée comme commutative avec elle-même. Ce qui est le cas de la déformation «Trouer» par exemple.

Le graphe de dépendances fonctionnelles n’intervient que lorsque des déformations sont réalisées sur une même région d’une surface. Il est utilisé par le moteur de planification. Afin de rester lisible, la figure 3.8 ne présente qu’une partie du graphe de dépendances fonctionnelles contenu dans la base de connaissances. Dans ce graphe, nous faisons le choix de dire que la déformation qui consiste à «bosseler» (Bo) une zone doit être réalisée dans tous les cas, après les déformations consistant à «Bomber» (Bb), «Creuser» (Cr) ou «Courber» (Cb) la surface et avant les opérations consistant à la trouser (Tr). Ce choix a été fait car nous considérons la déformation permettant de «bosseler» une zone comme étant une déformation de l’ordre de la finition. L’absence de déformation attachée à une zone est considérée comme une déformation vide notée «None» (No). Cette déformation vide apparaît dans le graphe de dépendances fonctionnelles de la base de connaissances comme n’ayant aucune influence sur les autres déformations.

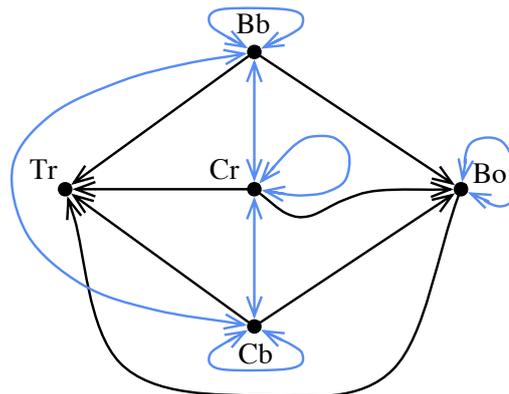


Figure 3.8. Extrait du graphe de dépendances fonctionnelles de la base de connaissances

Remarque 3.2.2 – Pour les traitements suivants, le graphe de dépendances fonctionnelles doit être construit de manière à ce que toutes les déformations soient en relation avec toutes les autres, en utilisant indifféremment des liens simples ou doubles. Ceci afin de connaître directement la relation d’ordre qui lie deux déformations. Seul le lien reliant une déformation à elle-même peut ne pas exister. Dans ce cas, la déformation est considérée comme commutative avec elle-même comme cela a été dit dans la remarque 3.2.1.

En plus de cette information d’ordre, la base de connaissances renferme la notion de *raffinement* attachée à chaque déformation. Cette donnée se présente sous la forme d’un coefficient reflétant le degré de raffinement que doit avoir la partie du polyèdre de contrôle attaché à une zone pour pouvoir être déformée. Ce concept est utilisé dans le processus de résolution/complément explicité dans le chapitre suivant.

3.2.3.2 Les localisations

La base de connaissances contient toutes les localisations disponibles dans le système déclaratif. Celles-ci permettent de caractériser les lieux et positions dans le plan paramétrique. Cela permet de placer les zones et donc de construire leur polygone de contrôle lors du processus de complétion numérique. Cette notion est la même que celle abordée dans le chapitre précédent lorsque l'on parle de référent absolu illustré sur la figure 2.10. Ce référent absolu est ici le référentiel R_i de la zone Z_i courante. Une localisation se différencie par des identifiants tels que «MOITIE GAUCHE», «ZONE CENTRALE», «ANGLE HG»... Un identifiant peut se décliner en plusieurs configurations qui reflètent les différentes façons de percevoir une même localisation. Cette notion est l'une des causes de l'obtention de plusieurs modèles sémantiques généralisés pour un seul modèle sémantique. En effet, si l'utilisateur parle d'«*angle haut-gauche*», il existe plusieurs façons de percevoir cette localité, nous en dissociions trois sur la figure 3.9. Durant la complétion sémantique, un modèle sémantique intermédiaire sera donc créé pour chaque configuration afin de proposer à l'utilisateur les trois solutions possibles. Une localisation se représente alors dans la base de connaissances comme une courbe NURBS fermée définie sur un espace plan $[0, 100]^2$ afin de pouvoir ensuite être directement mise en correspondance avec les référentiels R_i lors des différents traitements. La syntaxe XML utilisée est la même que pour la définition du polygone de contrôle d'une zone, les balises utilisées sont <Polygone> et <Moins>. La figure 3.9 illustre la définition de la localisation «ANGLE HG de Z_i » dans ses différentes configurations.

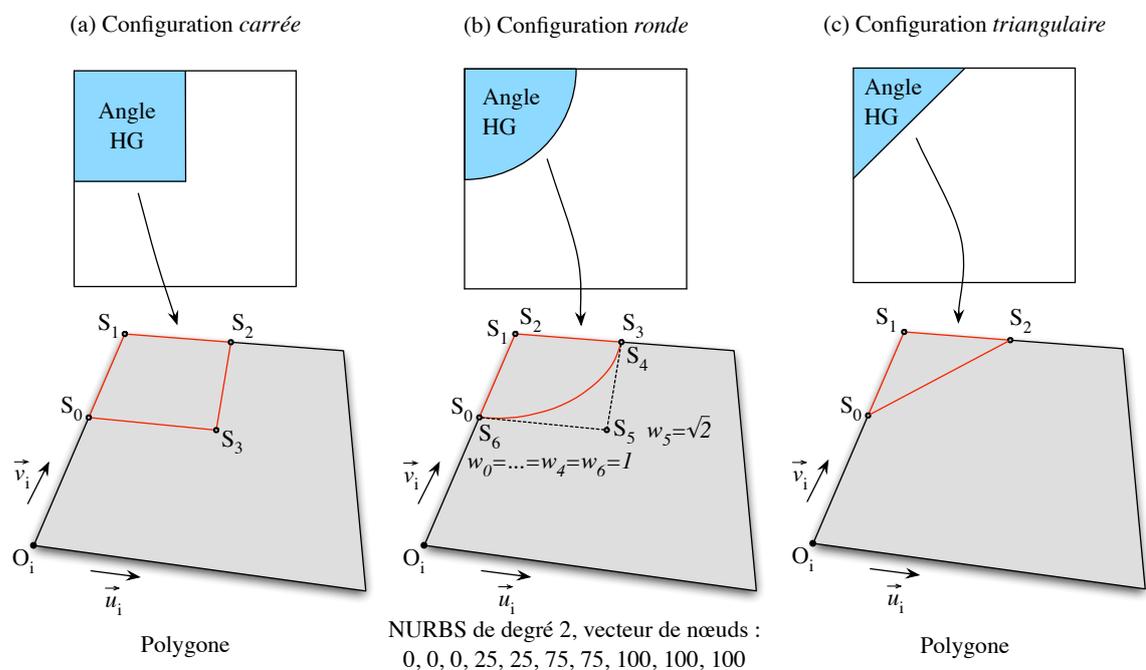


Figure 3.9. Localisation «ANGLE HG» et ses différentes configurations

La localisation que nous nommons «TOTALE» est une localisation particulière. En effet, elle correspond à la totalité de l'espace paramétrique disponible dans la zone mère de la zone courante. Ainsi une zone ayant la localisation totale a les mêmes frontières que sa zone mère.

3.2.3.3 Les formes prédéfinies

La liste des formes prédéfinies qui permettent de construire le polygone de contrôle de chaque zone est stockée dans la base de connaissances. Chaque forme est désignée par un identifiant qui lui est propre comme «CARREE», «RONDE»... La base de connaissances contient le polygone de contrôle de chaque forme que le système peut gérer. Notons que le principe est le même que pour \mathcal{P} , le degré de ce polygone de contrôle conditionne son traitement comme un polygone ou un polygone de contrôle. Celui-ci sert de patron dans le processus de complétion numérique pour construire le polygone de contrôle \mathcal{P} d'une zone. La syntaxe XML utilisée pour stocker ces polygones de contrôle est la même que pour la définition du polygone de contrôle d'une zone, ou d'une localisation. Elle utilise les balises <Polygone> et <Moins>. La figure 3.10 présente deux exemples de formes prédéfinies, la forme «CARREE» et la forme «RONDE».

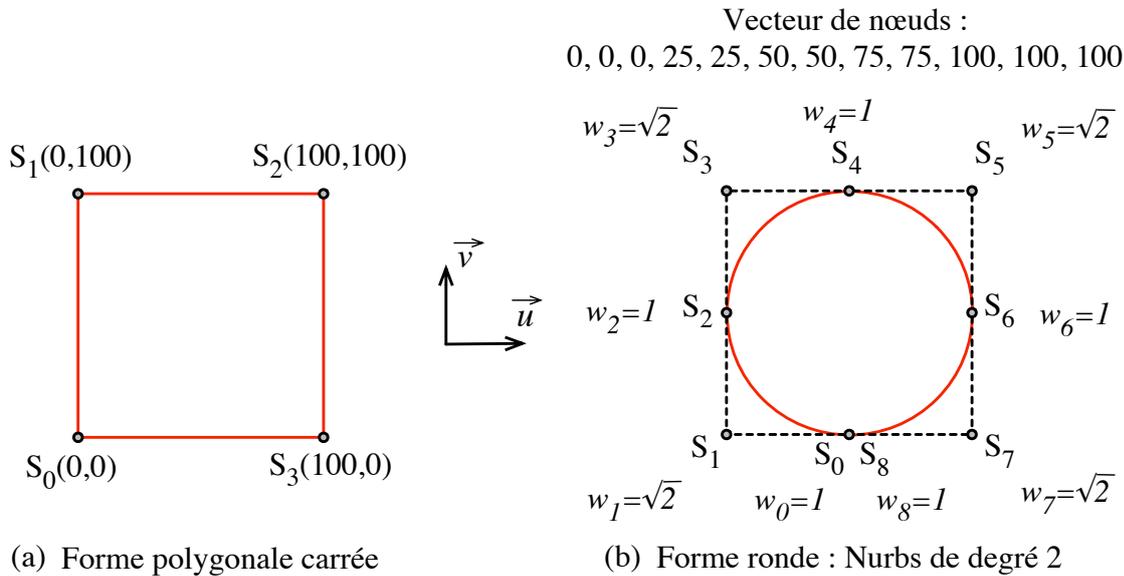


Figure 3.10. Polygones prédéfinis d'une forme «CARREE» (a) et «RONDE» (b)

Il existe beaucoup de façons de construire un cercle avec des courbes NURBS, la plus simple faisant intervenir 7 pôles en configuration de triangle comme le montre la figure 3.11. Nous choisissons une construction faisant intervenir 9 pôles sous forme d'un carré. Cette configuration a la propriété d'avoir son polyèdre de contrôle confondu avec la boîte englobante du cercle.

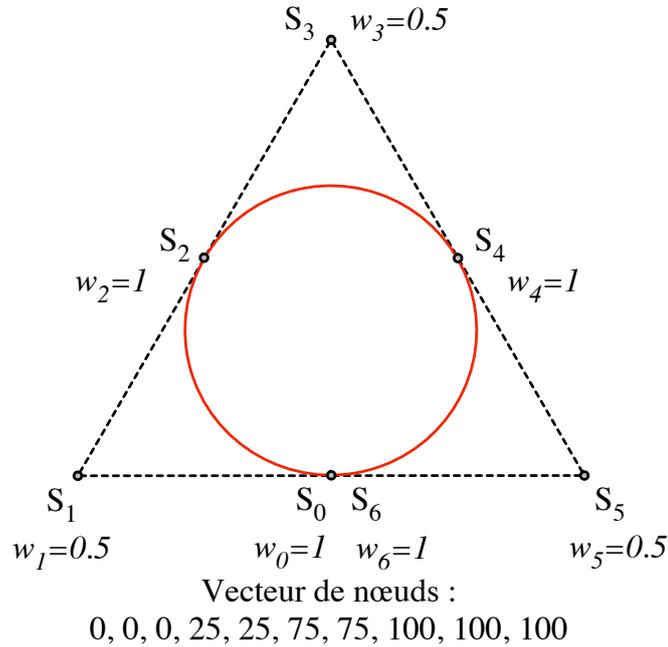


Figure 3.11. Courbe NURBS circulaire de degré 2 et définie par 7 pôles

Outre la définition par le polyèdre de contrôle, une forme peut s'exprimer en se servant d'une autre forme dite *base*, dont le polygone de contrôle est déjà connu. Ainsi une forme «RECTANGULAIRE» est construite sur la *base* du polygone d'une forme «CARREE» et une forme «OVALE» est construite sur la *base* du polygone de contrôle d'une forme «RONDE».

On attribue à ce type de forme des propriétés qui permettent de préciser leur aspect par rapport à leur forme de *base*. De cette manière, on distingue une forme «RECTANGULAIRE» qui peut être «LONGUE» d'une autre qui peut être «LARGE». Les formes qualifiées de *longues* (respectivement *larges*) ont leurs coordonnées selon l'axe \vec{u} (respectivement \vec{v}) doublées face à leurs coordonnées selon l'axe \vec{v} (respectivement \vec{u}). Une forme est ici définie parallèlement aux axes \vec{u} et \vec{v} des différents plans paramétriques locaux.

Nous utilisons deux identifiants particuliers de formes :

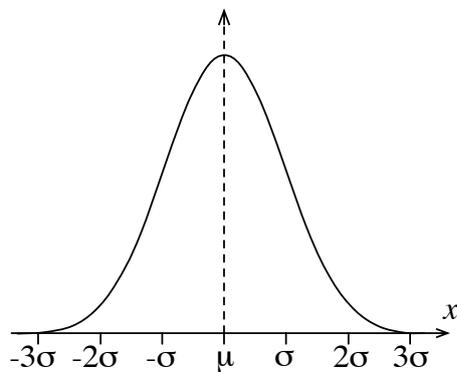
- «TOTALE» : une forme dite *totale* correspond exactement à la forme définie par sa localisation dans l'espace paramétrique local de sa zone mère. Si l'on prend l'exemple d'une zone Z_1 de forme *totale* localisée dans la partie droite de la zone Z_0 . Alors, Z_1 aura pour polygone de contrôle celui qui caractérise la localisation «PARTIE DROITE» défini sur R_0 et extrait de la base de connaissances ;
- «QUELCONQUE» : une forme dite quelconque n'a aucune contrainte particulière, ses frontières doivent simplement appartenir à la forme définie par sa localisation dans l'espace paramétrique local de sa zone mère.

3.2.3.4 Les quantificateurs

Enfin, à ce stade d'évolution du modèleur déclaratif de surfaces, la base de connaissances renferme les données permettant la gestion et l'interprétation des quantificateurs par une approche floue. Cette approche est apparue dès l'émergence du concept déclaratif et des propriétés dites *vagues* et *imprécises*. Les quantificateurs ont d'abord été numériquement représentés par des intervalles de valeurs et des fonctions trapézoïdales dans [Djedi, 1991]. Desmontils a aussi travaillé sur ce sujet [Desmontils, 1995a, Desmontils, 1996] et fait intervenir des opérateurs flous [Desmontils et Martin, 1997, Desmontils et Pacholczyk, 1997a, Desmontils et Pacholczyk, 1997b] pour aboutir dans [Desmontils, 2000] à une technique de résolution s'appuyant sur les CSP⁶.

Dans notre approche, chaque quantificateur est représenté par un identifiant qui le distingue des autres comme «*peu*» ou «*assez*». Cet identifiant est associé à une valeur comprise entre 0 et 100 qui permet de le positionner parmi les autres. Le pourcentage a été choisi comme moyen de quantification car son utilisation dans la vie courante en a fait un outil plus proche de l'utilisateur et plus intuitif à manipuler. Une valeur à 50 % est considérée comme *normale* ou *moyenne* alors qu'une valeur proche de 0 % est plutôt considérée comme faible et une valeur proche de 100 % est plutôt considérée comme élevée. Un quantificateur permet de contrôler la propriété à laquelle il est associé. Si l'on a par exemple : «*la zone centrale est assez bombée*», la propriété de déformation «*bombée*» associée à la zone centrale aura pour paramètre de contrôle la valeur attribuée au quantifiant «*assez*».

Afin de donner une *valeur* à un quantificateur tout en modélisant l'imprécision liée à l'approche déclarative, nous avons choisi d'assimiler chaque quantificateur à une variable aléatoire qui suit une loi de Laplace-Gauss [Saporta, 1996] (ou loi Normale) (eq. 3.1). La *valeur* précédemment citée est l'espérance⁷ (ou moyenne) μ à laquelle est associée un écart type σ qui permet de contrôler la portée du quantificateur. C'est cet écart type qui rend donc plus ou moins flou l'intervalle de valeurs possibles du quantificateur considéré.



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2} \quad (3.1)$$

⁶Constraint Satisfaction Problem

⁷valeur pour laquelle la probabilité est la plus forte

La figure 3.12 présente tous les quantificateurs disponibles dans notre approche. Chacun d'eux est ici associé à sa loi normale qui précise la probabilité des valeurs qu'il peut prendre. Ainsi le quantificateur «*assez*» peut prendre une valeur allant de $[\mu - 3\sigma; \mu + 3\sigma]$ ⁸ qui est 50 à 80, mais a une forte probabilité que celle-ci soit aux alentours de 66 qui est son espérance.

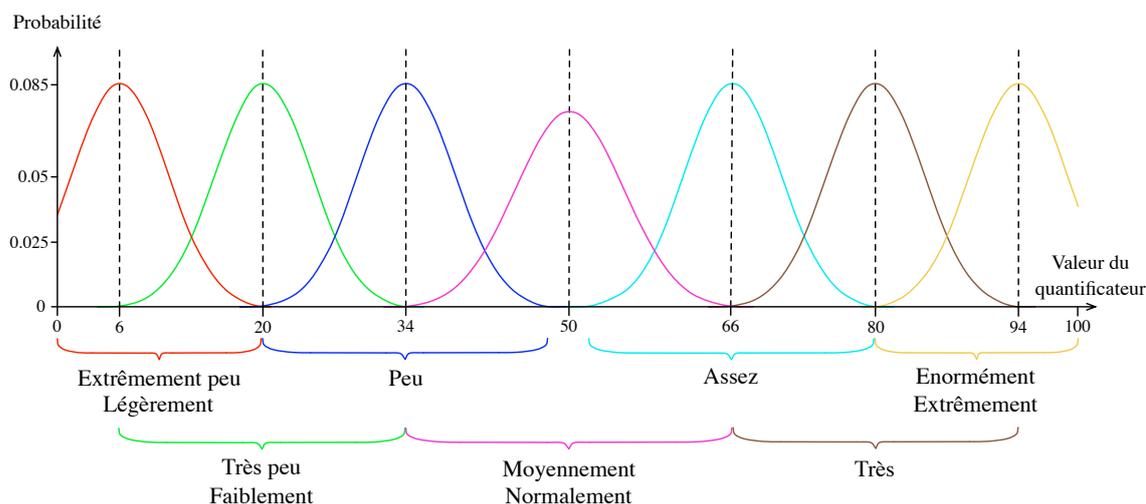


Figure 3.12. Graphique des probabilités et pourcentages attachés aux quantificateurs

Notre choix s'est porté sur une représentation avec recouvrement d'intervalles. Ceci afin d'avoir une certaine continuité entre les valeurs associées aux quantifiants mais aussi afin de renforcer la notion de frontières floues qui existe entre les termes et les identifiants utilisés. Le tableau 3.1 précise ces intervalles en donnant les espérances et les écarts types des quantificateurs exprimés sur la figure 3.12.

Identifiant	Espérance μ	Écart type σ	Écart type $\times 3$
<i>extrêmement peu</i>	6	4.67	14
<i>très peu</i>	20	4.67	14
<i>peu</i>	34	4.67	14
<i>normalement</i>	50	5.34	16
<i>assez</i>	66	4.67	14
<i>très</i>	80	4.67	14
<i>extrêmement</i>	94	4.67	14

TAB. 3.1. Identifiants et paramètres flous attachés aux quantificateurs

Dans la structure XML, ce sont les balises `<Quantificateur> ... </Quantificateur>` qui permettent d'exprimer la quantification. L'attribut «`Quantification="1"`» est ajouté à toute propriété, déformation ou localisation acceptant l'utilisation d'un quantificateur.

⁸Hors de cet intervalle, les probabilités sont tellement faibles, qu'elles sont ici considérées comme nulles

Cette base de connaissances est amenée à beaucoup évoluer avec les améliorations et les extensions du modèle déclaratif de surfaces. L'une des perspectives est notamment d'intégrer dans toute notre méthode des techniques d'apprentissage afin d'adapter la base de connaissances à l'utilisateur qui manipule le modèle déclaratif.

Fort de l'architecture des modèles sémantiques ainsi que des informations contenues dans la base de connaissances, nous abordons dans les parties suivantes les traitements qui y sont rattachés.

3.2.4 La complétion sémantique

La description donnée par l'utilisateur n'étant généralement pas assez précise, le modèle sémantique obtenu à l'issue du processus de traduction sémantique ne contient pas toutes les informations nécessaires au moteur de planification. La complétion sémantique permet alors de palier ce manque en déduisant certaines propriétés ou en complétant certains paramètres nécessaires. Ce processus est réalisé par le *moteur de complétion sémantique* dont on schématise l'architecture sur la figure 3.13. Celui-ci prend en entrée un unique modèle sémantique et peut en déduire plusieurs modèles sémantiques généralisés comme cela est expliqué dans ce paragraphe.

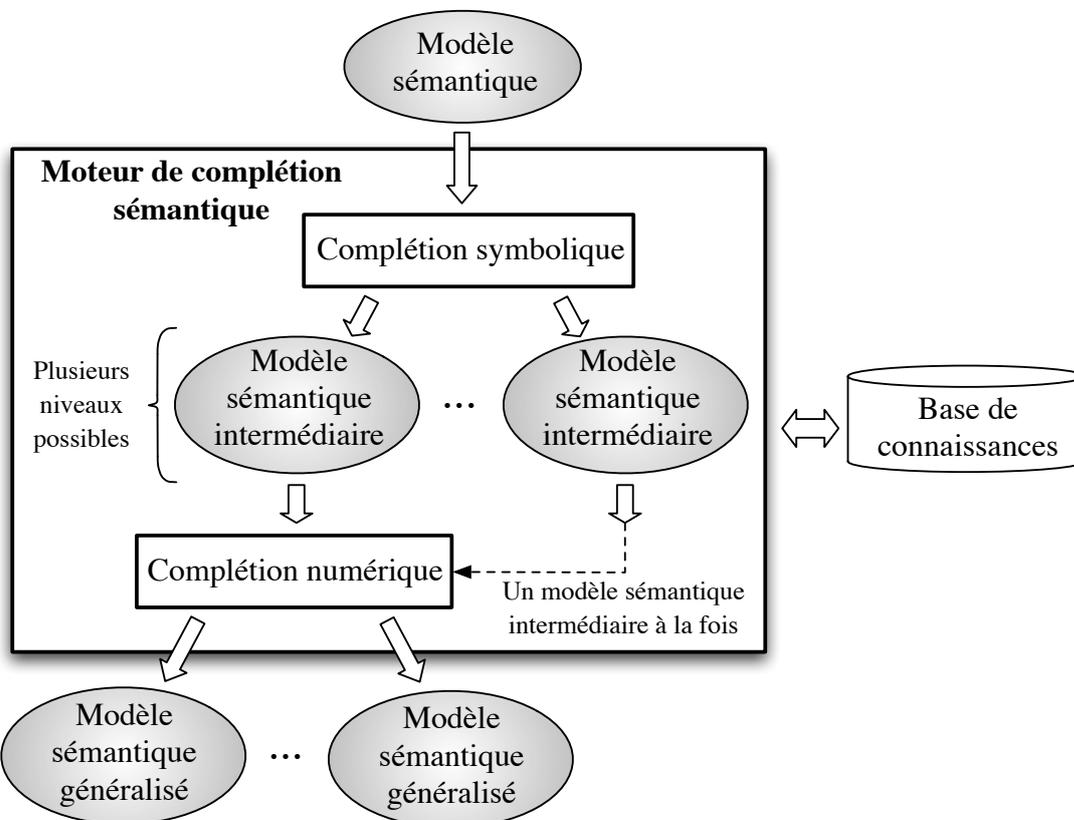


Figure 3.13. Du modèle sémantique vers ses généralisations : la complétion sémantique

Dans le cas où le modèle sémantique donné en entrée est déjà suffisamment riche en informations pour être traité par le processus de planification, le moteur de complétion sémantique ne réalise aucune opération. Il ne fait que transmettre le modèle sémantique qui est alors considéré comme déjà généralisé.

La complétion sémantique se réalise en deux étapes, la première consiste à compléter la symbolique du modèle, nous l'appelons *complétion symbolique*. La deuxième déduit de la symbolique les données numériques qui seront utilisées au cours des traitements suivants, nous la nommons *complétion numérique*. Les deux paragraphes suivants détaillent leur fonctionnement.

3.2.4.1 Complétion symbolique

Ce processus a pour but d'ajouter les informations symboliques manquantes au modèle sémantique. Comme nous l'avons précisé dans la partie décrivant la structure d'un modèle sémantique, un modèle sémantique généralisé doit impérativement contenir, pour chacune de ses zones, les informations suivantes :

- un polygone de contrôle défini à partir d'une forme et d'une localisation ;
- un ensemble de déformations ;
- un quantificateur pour toutes les propriétés et les déformations qui le permettent.

pour pouvoir être traité par le moteur de planification.

La technique que nous utilisons, consiste à détecter ces informations manquantes et à les ajouter au modèle courant. Ce procédé est itératif. Une itération consiste à appliquer la liste de règles présentée ci-dessous sur toutes les feuilles de l'arbre de solutions connues au début de cette itération. Au début du processus, cet arbre n'est constitué que de sa racine : le modèle sémantique à compléter qui est donné en entrée. Certaines règles peuvent être amenées à faire évoluer le modèle courant en d'autres modèles. Les nouveaux modèles seront alors considérés comme feuilles de l'arbre solution dès l'itération suivante. C'est ainsi, au fur et à mesure, que l'arbre solution croît. Les règles de complétion symbolique utilisées sont les suivantes :

1. La zone racine (Z_0) ne peut avoir qu'une localisation dite «*TOTALE*» (ceci afin de préciser que cette zone peut utiliser la totalité du plan paramétrique) ;
2. Si une zone a un *polygone de contrôle* défini, mais pas de *forme*, alors la forme associée à cette zone est considérée comme «*QUELCONQUE*» ;
3. Si une zone a un *polygone de contrôle* défini, mais pas de *localisation*, alors la localisation associée à cette zone est choisie dans la base de connaissances, en fonction de la position d'un ensemble de points régulièrement extraits du contour de cette zone ;
4. Si une propriété ou une localisation ou une déformation qui accepte un quantificateur n'en a pas un qui lui est associé, alors le quantificateur «*MOYENNEMENT*» ou «*NORMALEMENT*» lui est attaché ;
5. Si une zone n'a pas de *déformation*, alors la déformation «*NONE*» lui est attribuée ;

6. Si une zone n'a ni *forme*, ni *polygone de contrôle* définis, alors un modèle *sémantique intermédiaire* est créé pour chaque forme disponible dans la base de connaissances. Le modèle courant n'étant plus considéré comme feuille dans l'arbre des solutions, on arrête le traitement des règles pour ce modèle et on passe à une autre feuille de l'arbre des solutions ;
7. Si une zone n'a ni *localisation*, ni *polygone de contrôle* définis, alors la localisation associée à cette zone est «*TOTALE*» (cette zone peut alors utiliser la totalité de l'espace disponible dans sa zone mère) ;
8. Si une zone a une *localisation* mais n'a ni *forme*, ni *polygone de contrôle* définis, alors cette zone est associée à la forme dite «*TOTALE*». Ainsi ce sont les différentes configurations connues de sa localisation qui seront prises en compte comme forme ;
9. Si une zone a une *localisation* «ambiguë», alors un modèle *sémantique intermédiaire* est créé pour chaque configuration connue dans la base de connaissances associée à cette localisation. Le modèle courant n'étant plus considéré comme feuille dans l'arbre des solutions, on arrête le traitement des règles pour ce modèle et on passe à une autre feuille de l'arbre des solutions ;
10. Si une zone a une forme qui a besoin d'une propriété qualificative (comme une forme rectangulaire qui peut être «*longue*» ou «*large*») et que cette propriété n'est pas indiquée, alors un modèle *sémantique intermédiaire* est créé pour chaque propriété qualificative disponible dans la base de connaissances qui est attachée à la forme en question. Le modèle courant n'étant plus considéré comme feuille dans l'arbre des solutions, on arrête le traitement des règles pour ce modèle et on passe à une autre feuille de l'arbre des solutions ;

Ces règles sont appliquées sur toutes les feuilles de l'arbre des solutions jusqu'à ce qu'il n'y en ait plus aucune à vérifier. Ce processus itératif peut donc générer plusieurs modèles sémantiques intermédiaires pour un seul modèle sémantique donné en entrée. Chaque modèle sémantique intermédiaire considéré comme feuille de l'arbre des solutions est ensuite injecté dans le processus de complétion numérique expliqué dans le paragraphe suivant.

3.2.4.2 Complétion numérique

Les modèles intermédiaires intègrent maintenant la sémantique symbolique nécessaire. Le rôle de la complétion numérique est alors de passer de cette symbolique au numérique exploitable par le système. Nous proposons pour cela plusieurs règles à appliquer à toutes les zones de tous les modèles sémantiques intermédiaires feuilles de l'arbre des solutions :

1. Si un quantificateur n'a pas de valeurs numériques associées, alors il faut ajouter au modèle sémantique intermédiaire courant, les valeurs correspondant à l'identifiant du quantificateur, contenues dans la base de connaissances ;
2. Si une propriété n'a pas de valeurs numériques définies et qu'il en existe dans la base de connaissances, celles-ci sont ajoutées au modèle sémantique intermédiaire courant ;

3. Si le polygone de contrôle de la zone \mathcal{Z} n'est pas défini, alors il faut le construire en fonction de la forme et de la localisation de \mathcal{Z} . Cela passe par le positionnement de tous ses pôles ainsi que par la définition de son vecteur de nœuds et de son degré. Les pôles sont placés dans le repère défini par la boîte englobante de la zone mère de \mathcal{Z} . On fait ainsi intervenir pour la première fois la notion de plan paramétrique (u, v) qui sera ensuite utilisé par les modèles géométriques déduits du modèle sémantique généralisé en cours. Dans le cas de la construction du polygone de contrôle de \mathcal{Z} dont la localisation est absolue, les quatre cas suivants sont à prendre en compte :
 - (a) Si ni la forme, ni la localisation de \mathcal{Z} ne sont «TOTALE», alors la définition du polygone de contrôle de \mathcal{Z} consiste à placer le polygone de contrôle de sa forme, dont le patron est contenu dans la base de connaissances, dans la zone paramétrique définie par la courbe fermée de sa localisation ;
 - (b) Si \mathcal{Z} a sa forme que l'on qualifie de «TOTALE» et sa localisation définie comme non «TOTALE», alors \mathcal{Z} hérite du polygone de contrôle associé à sa localisation ;
 - (c) Si \mathcal{Z} a sa localisation que l'on qualifie de «TOTALE» et sa forme définie comme non «TOTALE», alors la définition du polygone de contrôle de \mathcal{Z} consiste à placer le polygone de contrôle de sa forme, dont le patron est contenu dans la base de connaissances, dans la zone paramétrique définie par la courbe fermée de sa zone mère (tout le repère local de \mathcal{Z}). Si \mathcal{Z} est la zone racine (\mathcal{Z}_0), les propriétés de forme (comme «longue» ou «large») ne sont pas prises en compte ici, elles le seront au niveau du processus de résolution/construction (Cf. 4.3) ;
 - (d) Si la forme et la localisation de \mathcal{Z} sont qualifiées de «TOTALE», alors \mathcal{Z} hérite du polygone de contrôle de sa zone mère définie dans le repère local de \mathcal{Z} .

Ce processus n'est pas itératif et a pour rôle de rendre exploitable les notions symboliques encore trop abstraites pour être directement utilisées dans les calculs des processus suivants. La troisième règle citée est essentielle pour toute la suite des traitements. La génération des polygones de contrôle des zones dont la localisation est absolue, est à traiter de façon prioritaire. La génération des polygones de contrôle des zones dont la localisation est relative, doit ensuite être traitée comme un problème de placement d'objets [Maculet, 1992] dans un espace borné à deux dimensions, cet espace étant ici le plan paramétrique local de la zone mère. La résolution de ce positionnement est loin d'être triviale et la combinatoire reste parfois très élevée. Nous n'avons pas souhaité traiter ce problème dans le cadre de cette thèse et le réservons aux perspectives d'évolution de notre approche. Cependant, le positionnement des sommets des polygones de contrôle des différentes zones peut être résolu par des méthodes utilisant des CSP en discrétisant le plan paramétrique courant [Lauriere, 1976, Maculet, 1991] ou bien en utilisant des CSP numériques [Lhomme et Rueher, 1997, Kwaiter and al., 1997, Delobel, 2000, Jermann, 2002] en travaillant sur des intervalles paramétriques. Des méthodes basées sur des techniques de configuration [Henocque et al., 2005] peuvent aussi être appliquées dans l'espace à deux

dimensions. Des travaux s’orientant vers cette dernière méthode et s’intégrant à ceux déjà menés dans le cadre de la traduction sémantique de description (Cf. 3.2) sont d’ailleurs actuellement en cours dans l’équipe INCA du LSIS.

Afin de mieux comprendre l’architecture des connaissances et l’évolution des modèles au cours de la complétion sémantique, la partie suivante présente un exemple simple du passage d’une description vers l’ensemble de ses modèles sémantiques généralisés.

3.2.4.3 Application sur un exemple simple

Soit la description suivante fournie par l’utilisateur comme exemple 1 :

« *La moitié droite de la surface est bombée ;
La partie basse de la surface est peu creusée* ».

Le modèle sémantique obtenu après traduction sémantique pourrait être :

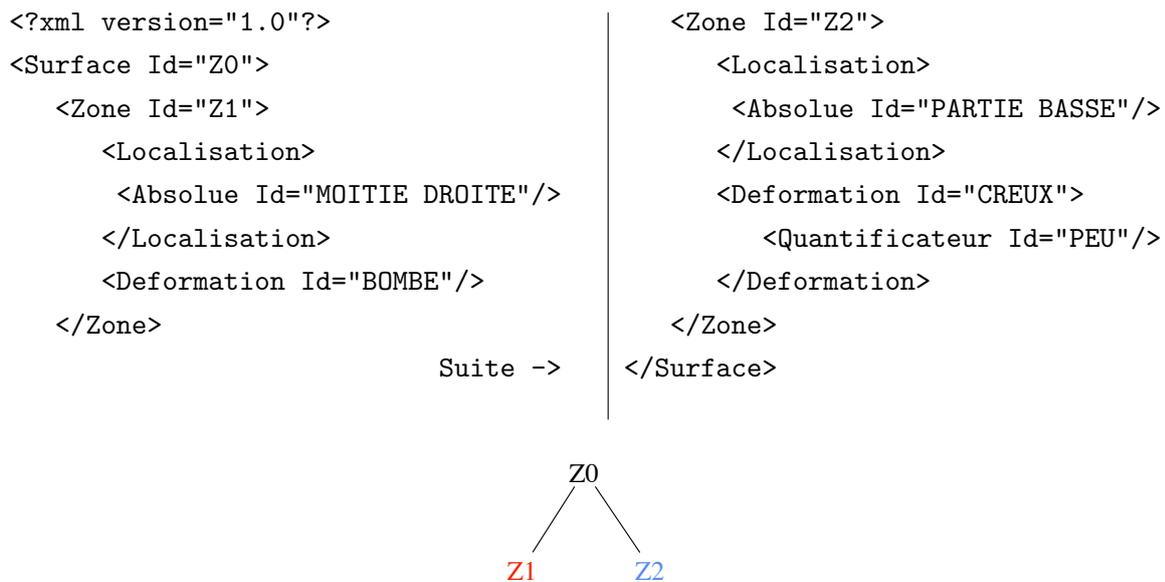


Figure 3.14. Ex. 1 : Arbre structurant

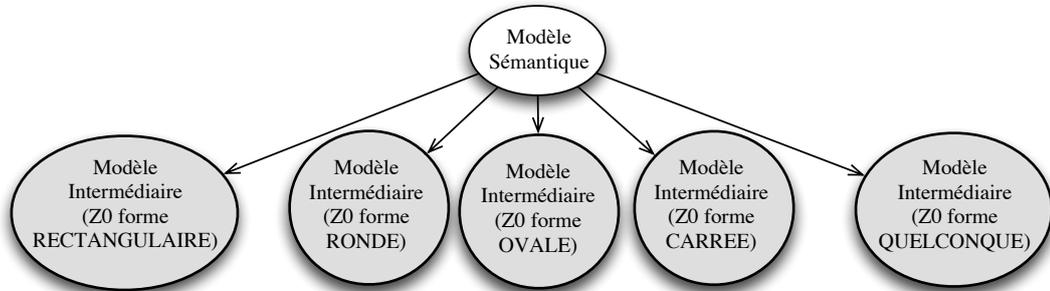
La figure 3.14 présente l’arbre structurant rattaché à la description donnée. Seules trois zones interviennent dans la description ce qui rend la structure présentée très simple. Le modèle sémantique produit pourrait être issu d’un module d’interfaçage destiné à traiter des descriptions écrites en langage naturel, ou pseudo naturel, basé sur la langue française. Ce modèle n’intègre que les informations extraites de la description lors du processus de traduction. Il est ensuite injecté dans le moteur de complétion sémantique où il est d’abord traité sur le plan *symbolique*. Ce processus consiste à appliquer les règles évoquées dans la partie 3.2.4.1 de façon itérative sur chacune des zones figurant dans les modèles feuilles de l’arbre des solutions. Ces modèles sont représentés sur fond gris.

Avant le processus de compléssion sémantique, l'arbre des solutions se réduit au modèle sémantique :

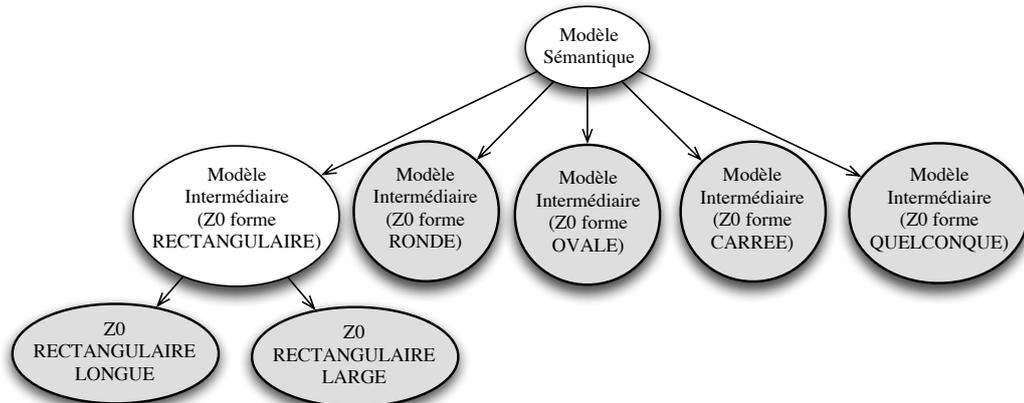


Sur le modèle sémantique, on commence par la zone Z0 :

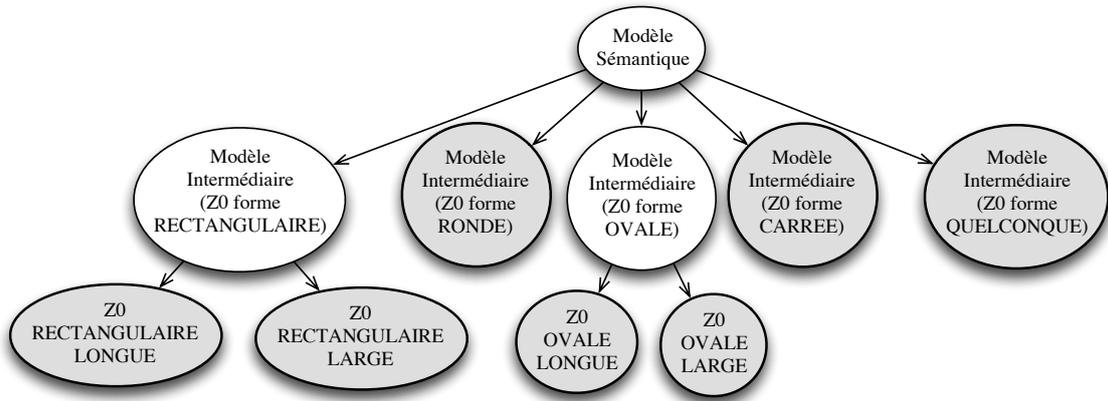
- Itération 1 :
 - Règle 1 : on lui associe la localisation dite «TOTALE» ;
 - Règle 5 : on lui associe la déformation *vide* dite «NONE» ;
 - Règle 6 : Le modèle sémantique courant évolue en 5 modèles sémantiques intermédiaires intégrant chacun une des formes suivantes : RECTANGULAIRE, RONDE, CARREE, OVALE et QUELCONQUE. Le modèle sémantique était la seule feuille au début du processus, on passe à l'itération 2 qui prendra en compte les nouveaux modèles.



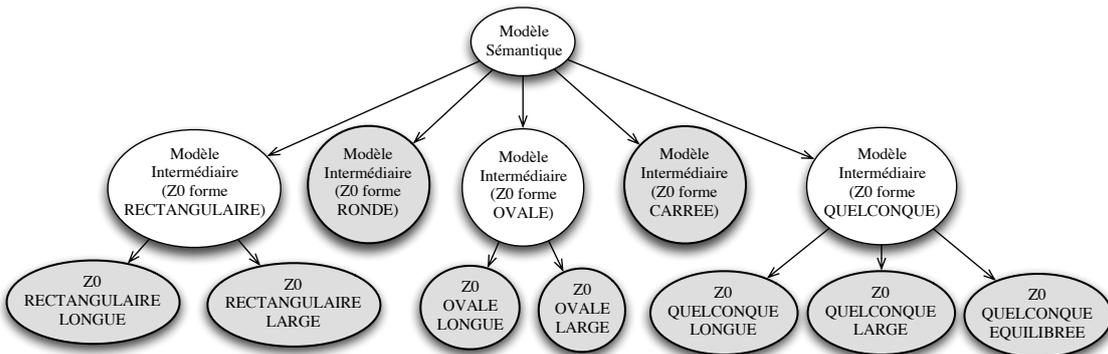
- Itération 2 :
 - Sur le modèle intermédiaire (Z0 forme RECTANGULAIRE) :
 - Règle 10 : on fait évoluer le modèle sémantique intermédiaire courant en 2 nouveaux modèles sémantiques intermédiaires intégrant chacun une des propriétés de la forme «RECTANGULAIRE» : LONGUE ou LARGE.



- Sur le modèle intermédiaire (Z0 forme OVALE) :
 - Règle 10 : on fait évoluer le modèle sémantique intermédiaire courant en 2 nouveaux modèles sémantiques intermédiaires intégrant chacun une des propriétés de la forme «RECTANGULAIRE» : LONGUE ou LARGE.



- Sur le modèle intermédiaire (Z0 forme QUELCONQUE) :
 - Règle 10 : on fait évoluer le modèle sémantique intermédiaire courant en 2 nouveaux modèles sémantiques intermédiaires intégrant chacun une des propriétés de la forme «RECTANGULAIRE» : LONGUE, LARGE ou EQUILIBREE.



- Itération 3 :
 - Sur les 6 modèles intermédiaires issus des 7 générés à l'itération 2 contenant les propriétés «LONGUE» et «LARGE» :
 - Règle 4 : on ajoute aux propriétés «LONGUE», «LARGE» le quantificateur «NORMALEMENT».
- Itération 4 : aucune règle concernant Z0 ne s'applique plus sur les 9 modèles intermédiaires considérés comme feuille dans l'arbre des solutions.

Après le traitement de Z0 qui a été réalisé en quatre itérations, l'arbre des solutions est constitué de 9 modèles sémantiques intermédiaires considérés comme feuilles de l'arbre des solutions. La figure 3.15 représente ces 9 modèles sémantiques intermédiaires sur fond gris. Le traitement des zones suivantes, Z1 puis Z2, s'applique sur ces 9 derniers modèles.

À titre d'exemple voici le code XML du modèle sémantique intermédiaire «Z0 à forme RECTANGULAIRE LONGUE» à ce stade du processus :

```

<?xml version="1.0"?>
<Surface Id="Z0">
  <Localisation>
    <Absolue Id="TOTALE"/>
  </Localisation>
  <Forme Id="RECTANGULAIRE">
    <Propriete Id="LONGUE">
      <Quantificateur Id="MOYENNEMENT"/>
    </Propriete>
  </Forme>
  <Deformation Id="NONE"/>
  <Zone Id="Z1">
    <Localisation>
      <Absolue Id="MOITIE DROITE"/>
    </Localisation>
    <Deformation Id="BOMBE"/>
  </Zone>
  <Zone Id="Z2">
    <Localisation>
      <Absolue Id="PARTIE BASSE"/>
    </Localisation>
    <Deformation Id="CREUX">
      <Quantificateur Id="PEU"/>
    </Deformation>
  </Zone>
</Surface>
Suite ->

```

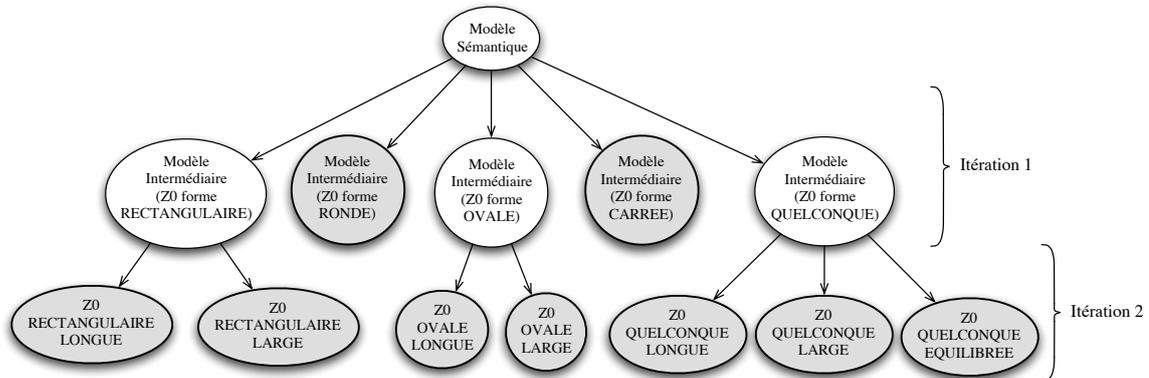


Figure 3.15. Ex. 1 : Arbre des solutions après la complétion symbolique de Z0

Sur la zone Z1 :

- Itération 1 :
 - Sur tous les modèles intermédiaires feuilles de l'arbre des solutions :
 - Règle 4 : on ajoute le quantificateur «NORMALEMENT» à la déformation «BOMBE» ;
 - Règle 8 : on lui ajoute la forme dite «TOTALE» ;
- Itération 2 : aucune règle concernant Z1 ne s'applique plus sur les 9 modèles intermédiaires considérés comme feuilles dans l'arbre des solutions.

Sur la zone Z2 :

- Itération 1 :
 - Sur tous les modèles intermédiaires feuilles de l'arbre des solutions :
 - Règle 8 : on lui ajoute la forme dite «TOTALE» ;
- Itération 2 : aucune règle concernant Z1 ne s'applique plus sur les 9 modèles intermédiaires considérés comme feuille dans l'arbre des solutions.

Les 9 modèles sémantiques intermédiaires considérés comme feuilles de l'arbre des solutions contiennent maintenant toute la sémantique symbolique nécessaire. Ils peuvent être injectés un à un dans le moteur de *complétion numérique* où les règles citées dans la partie 3.2.4.2 leur sont appliquées. Nous ne traitons ici que le cas du modèle sémantique intermédiaire identifié par «Z0 à forme RECTANGULAIRE LONGUE» dont voici à titre d'exemple le code XML à la fin du processus de complétion symbolique :

<pre> <?xml version="1.0"?> <Surface Id="Z0"> <Localisation> <Absolue Id="TOTALE"/> </Localisation> <Forme Id="RECTANGULAIRE"> <Propriete Id="LONGUE"> <Quantificateur Id="MOYENNEMENT"/> </Propriete> </Forme> <Deformation Id="NONE"/> <Zone Id="Z1"> <Localisation> <Absolue Id="MOITIE DROITE"/> </Localisation> </pre>	<pre> <Forme Id="TOTALE"/> <Deformation Id="BOMBE"> <Quantificateur Id="MOYENNEMENT"/> </Deformation> </Zone> <Zone Id="Z2"> <Localisation> <Absolue Id="PARTIE BASSE"/> </Localisation> <Forme Id="TOTALE"/> <Deformation Id="CREUX"> <Quantificateur Id="PEU"/> </Deformation> </Zone> </Surface> </pre>
---	--

Suite ->

Appliquons maintenant le processus de complétion numérique sur ce modèle sémantique intermédiaire :

Sur le modèle sémantique intermédiaire (Z0 RECTANGULAIRE LONGUE) :

- Zone Z0 :
 - Règle 1 : les données numériques issues de la base de connaissances sont ajoutées au quantificateur «MOYENNEMENT» de la propriété «LONGUE» ;
 - Règle 2 : les données numériques issues de la base de connaissances sont ajoutées à la propriété «LONGUE» ;
 - Règle 3.(c) : le polygone de contrôle patron contenu dans la base de connaissances et associé à la forme rectangulaire est placé dans le plan paramétrique du modèle sémantique intermédiaire courant. les quatre sommets du polygone de contrôle sont placés aux quatre extrémités du plan paramétrique. La zone considérée étant la zone racine, la propriété précisant que la forme est longue n'est pas prise en compte à ce niveau ;
- Zone Z1 :
 - Règle 1 : les données numériques issues de la base de connaissances sont ajoutées au quantificateur «MOYENNEMENT» de la déformation «BOMBE» ;

- Règle 3.(b) : on ajoute directement le polygone de contrôle contenu dans la base de connaissances et associé à la localisation «moitié droite» comme polygone de contrôle de Z_1 ;
- Zone Z_2 :
 - Règle 1 : les données numériques issues de la base de connaissances sont ajoutées au quantificateur «PEU» de la déformation «CREUX» ;
 - Règle 3.(b) : on ajoute le polygone de contrôle défini dans la base de connaissances, associé à la localisation «partie basse» directement comme polygone de contrôle de Z_2 .

Le principe est identique pour les 8 autres modèles sémantiques intermédiaires. À la fin de la complétion numérique, tous les modèles intermédiaires considérés comme feuilles de l'arbre des solutions gagnent leur rang de modèle sémantique généralisé. Voici en exemple le code XML de celui issu du modèle sémantique intermédiaire identifié par « Z_0 à forme RECTANGULAIRE LONGUE» :

<pre> <?xml version="1.0"?> <!DOCTYPE Surface SYSTEM "ModeleSemantique.dtd"> <Surface Id="Z0"> <Localisation> <Absolue Id="TOTALE"/> </Localisation> <Forme Id="RECTANGULAIRE"> <Propriete Id="LONGUE" U="1.0000" V="2.0000"> <Quantificateur Id="MOYENNEMENT" Ecart="5.34" Valeur="50.0000"/> </Propriete> </Forme> <Deformation Id="NONE"/> <Polygone> <Sommet Id="S0" U="0" V="0"/> <Sommet Id="S1" U="0" V="100"/> <Sommet Id="S2" U="100" V="100"/> <Sommet Id="S3" U="100" V="0"/> </Polygone> <Zone Id="Z1"> <Localisation> <Absolue Id="MOITIE DROITE"/> </Localisation> <Forme Id="TOTALE"/> <Deformation Id="BOMBE"> <Quantificateur Id="MOYENNEMENT" Ecart="5.34" Valeur="50.0000"/> </Deformation> </Zone> <Polygone> <Sommet Id="S0" U="50" V="0"/> <Sommet Id="S1" U="50" V="100"/> <Sommet Id="S2" U="100" V="100"/> <Sommet Id="S3" U="100" V="0"/> </Polygone> </Zone> </Surface> </pre>	<pre> <Polygone> <Sommet Id="S0" U="50" V="0"/> <Sommet Id="S1" U="50" V="100"/> <Sommet Id="S2" U="100" V="100"/> <Sommet Id="S3" U="100" V="0"/> </Polygone> </Zone> <Zone Id="Z2"> <Localisation> <Absolue Id="PARTIE BASSE"/> </Localisation> <Forme Id="TOTALE"/> <Deformation Id="CREUX"> <Quantificateur Id="PEU" Ecart="4.67" Valeur="34.0000"/> </Deformation> <Polygone> <Sommet Id="S0" U="0" V="0"/> <Sommet Id="S1" U="0" V="50"/> <Sommet Id="S2" U="100" V="50"/> <Sommet Id="S3" U="100" V="0"/> </Polygone> </Zone> </Surface> </pre>
---	--

Suite ->

Chaque modèle sémantique généralisé ainsi obtenu est injecté dans le moteur de planification dont nous détaillons le fonctionnement dans la partie suivante.

3.3 Le moteur de planification

3.3.1 Notion de classe de solutions : le modèle patron

Un modèle sémantique généralisé définit un ensemble de zones sur un plan paramétrique. L'agencement de ces zones répond à la description donnée par l'utilisateur. Chaque zone est rattachée à un ensemble de déformations et à un polygone de contrôle qui précise ses bords dans le plan paramétrique. Il faut donc appliquer toutes les déformations de chacune des zones du modèle sémantique généralisé pour obtenir une forme potentiellement solution pour l'utilisateur. La réalisation de ces déformations reste cependant une tâche délicate lorsque deux zones paramétriques se chevauchent ou qu'une zone a plus d'une déformation qui lui est associée. Dans ces cas, la forme résultat dépend directement de l'ordre d'application des déformations. L'opération de déformation n'étant, dans la majorité des cas, pas commutative, une simple inversion peut engendrer des résultats différents.

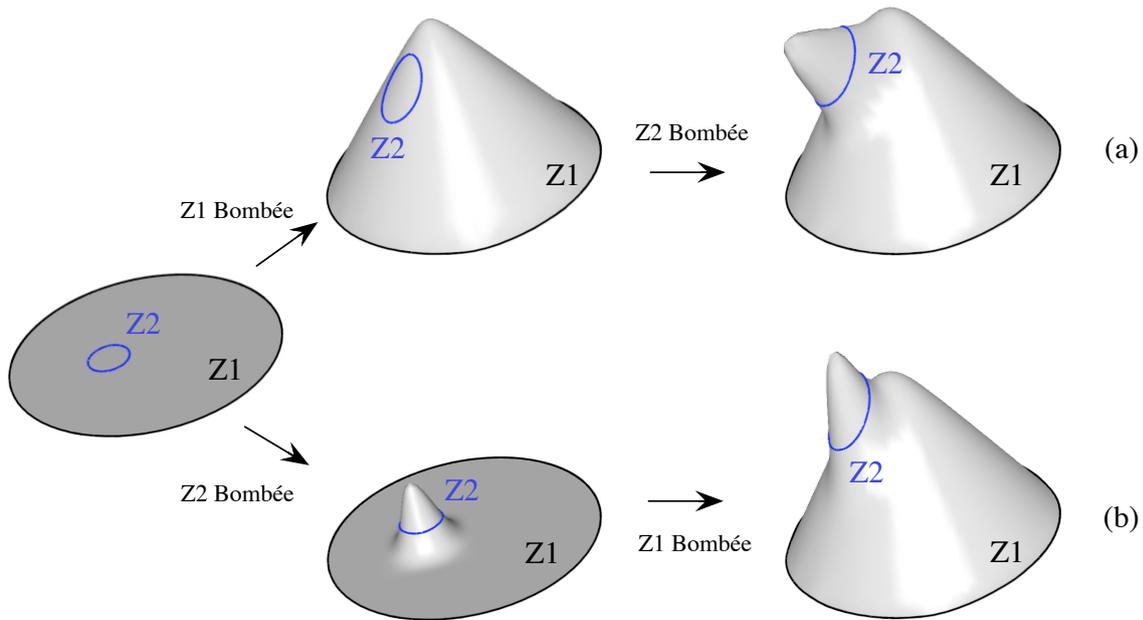


Figure 3.16. Non commutativité des déformations

La figure 3.16 présente une superposition de deux zones, $Z1$ et $Z2$, de forme ronde. Chacune de ces zones est rattachée à une déformation de type «Bombé». En partant d'un support de surface plan, si on applique la déformation de $Z1$ puis celle de $Z2$ on obtient par exemple la forme résultat figurant sur la branche (a) de la figure. Si, par contre, c'est la déformation de $Z2$ qui est appliquée avant celle de $Z1$, on obtient plutôt le résultat figurant sur la branche (b).

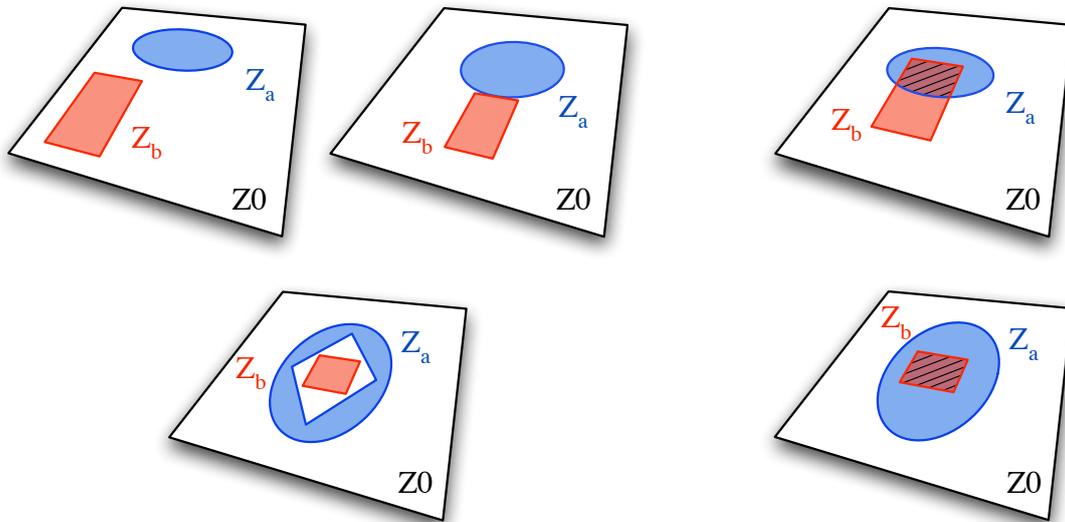
Le résultat d'une déformation est aussi, la plupart du temps, conditionné par la forme de la surface avant cette déformation. La forme initiale de la surface pouvant elle-même être considérée comme le résultat d'une première déformation. L'ordre des manipulations de la

surface est ainsi très important puisque chaque modification du modèle géométrique peut influencer les résultats suivants si les déformations se chevauchent. Afin de tenir compte de ce phénomène d'influence entre déformations, nous nous reportons aux zones auxquelles elles sont définies pour introduire les notions d'*indépendance* et de *dépendance mutuelle* entre zone dont voici nos définitions :

Définition 3.3.1 – Deux zones Z_1 et Z_2 sont dites en *dépendance mutuelle* lorsque les régions que définissent leurs frontières se chevauchent. On dit aussi que Z_1 est en *recouvrement* avec Z_2 ou que Z_2 est en *recouvrement* avec Z_1 . Les zones tangentes ne sont pas considérées comme étant en *dépendance mutuelle*.

Définition 3.3.2 – Deux zones Z_1 et Z_2 sont considérées comme *indépendantes* lorsque les régions que définissent leurs frontières ne se chevauchent pas, c'est-à-dire lorsqu'elles ne sont pas en *dépendance mutuelle*. On dit aussi qu'il y a *non recouvrement* entre Z_1 et Z_2 . Les zones tangentes sont considérées comme étant *indépendantes*.

La figure 3.17 présente les différents cas de recouvrement et de non-recouvrement possibles entre deux zones Z_a et Z_b .



(a) Pas de chevauchement entre Z_a et Z_b

(b) Chevauchement entre Z_a et Z_b

Figure 3.17. Étape 2 : prise en compte des chevauchements entre zones

La méthode générale que nous utilisons pour détecter les recouvrements entre zones se réalise en deux temps :

1. Détection des couples de zones (Z_a, Z_b) en chevauchement potentiel. Le premier test se fait au niveau des boîtes englobantes de Z_a et Z_b : si un seul sommet de la boîte englobante de Z_a est strictement inclus dans la boîte englobante de Z_b ou qu'un seul

sommet de la boîte englobante de \mathcal{Z}_b est strictement inclus dans la boîte englobante de \mathcal{Z}_a alors ce couple de zones passe à l'étape (b) sinon c'est qu'il n'y a pas de recouvrement entre les deux zones. Cette première étape, très classique, permet donc de rapidement éliminer les couples de zones indépendantes ;

2. Vérification d'un chevauchement potentiel entre \mathcal{Z}_a et \mathcal{Z}_b : le but n'étant pas de déterminer le lieu de recouvrement exact, nous considérons une courbe NURBS fermée de degré strictement supérieur à 1 comme un polygone en la discrétisant suffisamment pour être très proche de sa représentation continue. Si le degré est égal à 1, nous rappelons que les frontières de la zone sont directement \mathcal{P} considérées comme un polygone. La méthode citée en (a) n'est alors plus appliquée sur les boîtes englobantes mais directement sur les polygones et les versions discrètes des courbes NURBS fermées. \mathcal{Z}_a est ainsi considérée en recouvrement avec \mathcal{Z}_b si l'un des sommets de \mathcal{Z}_a est strictement inclus dans \mathcal{Z}_b ou si l'un des sommets de \mathcal{Z}_b est strictement inclus dans \mathcal{Z}_a .

Remarquons, par ailleurs, que l'étape 1 n'est pas nécessaire pour obtenir un résultat correct, seule l'étape 2 permet de conclure. L'étape 1 permet simplement de gagner un grand temps de calcul pour des polygones dont le nombre de sommets est élevé (cas des courbes NURBS discrétisées par exemple). Ainsi, pour des polygones dont le nombre de sommets est faible, seule l'étape 2 peut être appliquée.

La *dépendance mutuelle* ou bien au contraire, l'*indépendance* entre zones, conditionnent complètement le résultat obtenu lors des déformations. En effet, l'ordre des déformations n'est pas important si les zones sont indépendantes, alors qu'il devient indispensable lorsqu'il existe au moins une dépendance mutuelle. Ainsi, dans le but de proposer à l'utilisateur le plus grand nombre possible de solutions différentes, tout en tenant compte de l'ordonnement des déformations, nous introduisons les notions de *séquence de construction* et de *classe de solutions*, dont voici nos définitions :

Définition 3.3.3 – Une **séquence de construction** \mathcal{S}_i est un ordre possible d'application de déformations. Une séquence se présente comme une chaîne de couples (Zone, Déformation) ou chaque couple est noté ZD . On parle aussi d'opération puisque ZD peut se lire : appliquer la déformation D sur la Zone \mathcal{Z} .

Deux séquences sont dites égales si elles sont composées des mêmes couples dans le même ordre. Deux séquences $ZD1$ et $ZD2$ sont dites équivalentes si les modèles géométriques obtenus par l'application des mêmes déformations selon $ZD1$ ou $ZD2$ sont strictement identiques. Ainsi des séquences impliquant des zones indépendantes sont forcément équivalentes. On définit la dimension d'une séquence comme le nombre de couples qui la constituent.

Exemple : Soit trois zones \mathcal{Z}_1 , \mathcal{Z}_2 et \mathcal{Z}_3 avec \mathcal{Z}_1 qui est bombée, \mathcal{Z}_2 qui est creusée et \mathcal{Z}_3 qui est courbée. On admet qu'il existe un recouvrement entre \mathcal{Z}_1 et \mathcal{Z}_2 , qu'il n'existe pas de relation hiérarchique entre les zones et qu'il n'existe pas de contrainte d'ordre sur les déformations.

Pour cet exemple, il existe donc 3 couples (*Zone, Déformation*) qui sont : (Z_1, Bb) , (Z_2, Cr) et (Z_3, Cb) . Ce qui correspond à 6 ordres d'applications possibles puisque c'est le nombre de combinaisons de 3 éléments, soit $3! = 6$. Ainsi, 6 séquences de construction sont rattachées à cet exemple telles que :

$$ZD0 = (Z_1, Bb) (Z_2, Cr) (Z_3, Cb);$$

$$ZD1 = (Z_1, Bb) (Z_3, Cb) (Z_2, Cr);$$

$$ZD2 = (Z_2, Cr) (Z_3, Cb) (Z_1, Bb);$$

$$ZD3 = (Z_2, Cr) (Z_1, Bb) (Z_3, Cb);$$

$$ZD4 = (Z_3, Cb) (Z_1, Bb) (Z_2, Cr);$$

$$ZD5 = (Z_3, Cb) (Z_2, Cr) (Z_1, Bb);$$

Dans cet ensemble, aucune de ces séquences n'est égale. Cependant, Z_3 est indépendante de Z_1 et Z_2 , donc l'application de sa déformation n'a pas d'impact sur l'ordre de la séquence. On peut ainsi dire que les séquences dont l'ordre de Z_2 et Z_3 sont les mêmes sont équivalentes. On a donc : $ZD0$, $ZD2$ et $ZD3$ qui sont équivalentes et $ZD1$, $ZD4$ et $ZD5$ qui sont équivalentes.

Nous définissons ces ensembles de séquences équivalentes comme étant des classes de solutions telles que :

Définition 3.3.4 – *Une classe de solutions est l'ensemble des modèles géométriques de surfaces issues d'un même modèle sémantique généralisé et partageant des séquences de construction équivalentes. Un tel ensemble est représenté par un **modèle patron** défini par une des séquences de construction équivalentes appartenant à la classe en question.*

Exemple : Si l'on reprend les résultats de l'exemple précédent, il existe dans ce cas deux classes de solutions qui sont : $ZD0$, $ZD2$, $ZD3$ et $ZD1$, $ZD4$, $ZD5$. On peut donc identifier deux modèles patrons, l'un défini par $ZD0$ et l'autre par $ZD1$ par exemple, qui représente chacun une classe de solutions.

Pour déterminer toutes les classes de solutions d'un même modèle sémantique généralisé, nous avons mis en place un moteur de *planification*. Celui-ci permet de déterminer toutes les différentes séquences de construction possibles de déformations et de les regrouper en classes d'équivalence. En plus de la notion de recouvrement de zones que nous venons de définir, ce moteur tient compte de la hiérarchie des zones en ajoutant la contrainte que les déformations de la zone mère doivent forcément être réalisées avant celles de ses zones filles. Mais il tient aussi compte de la dépendance des déformations définies dans la base de connaissances à travers le graphe de dépendances fonctionnelles, ajoutant des contraintes d'ordre selon le type de déformations appliquées. La technique utilisée dans ce processus est détaillée dans la partie suivante.

3.3.2 Planification à partir d'un modèle sémantique généralisé

Le moteur de planification prend en entrée un modèle sémantique généralisé et a pour objectif de déduire tous les modèles patrons (ou classes de solutions) qui y sont associés. Ce procédé que nous avons développé, se déroule en deux phases comme le montre la figure 3.18.

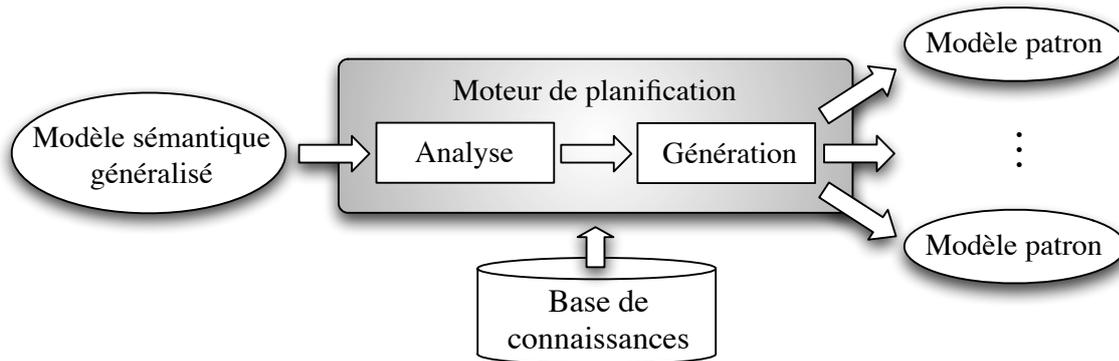


Figure 3.18. Moteur de planification

La première phase dite d'*analyse* du modèle sémantique généralisé, regroupe et structure les données hiérarchiques de ce modèle en y intégrant les dépendances mutuelles entre zones et les dépendances entre déformations. La structure finale obtenue est un graphe que l'on nomme *graphe de dépendances inter-zones* noté \mathcal{G}_{diz} . Celui-ci se construit en deux temps :

- une première étape consiste à réunir les informations hiérarchiques et les dépendances mutuelles entre zones en construisant un graphe intermédiaire nommé *graphe de dépendances spatiales et hiérarchiques* et noté \mathcal{G}_{dsh} ;
- La seconde étape consiste à intégrer à \mathcal{G}_{dsh} , les contraintes issues des dépendances entre déformations contenues dans la base de connaissances sous la forme du graphe de dépendances fonctionnelles \mathcal{G}_{df} . Le graphe obtenu est alors le *graphe de dépendances inter-zones* \mathcal{G}_{diz} .

La deuxième phase dite de *génération* interprète \mathcal{G}_{diz} pour en déduire les séquences de construction possibles regroupées en classes d'équivalence. Ces ensembles de séquences équivalentes représentent les classes de solutions attachées au modèle sémantique généralisé et permettent ainsi de déduire ses modèles patrons. Les différentes séquences peuvent aussi se représenter sous la forme d'un graphe que l'on nomme *graphe de séquences de construction* noté \mathcal{G}_{sc} .

La figure 3.19 schématise cet enchaînement de constructions de graphes avant de détailler les méthodes utilisées pour ces opérations.

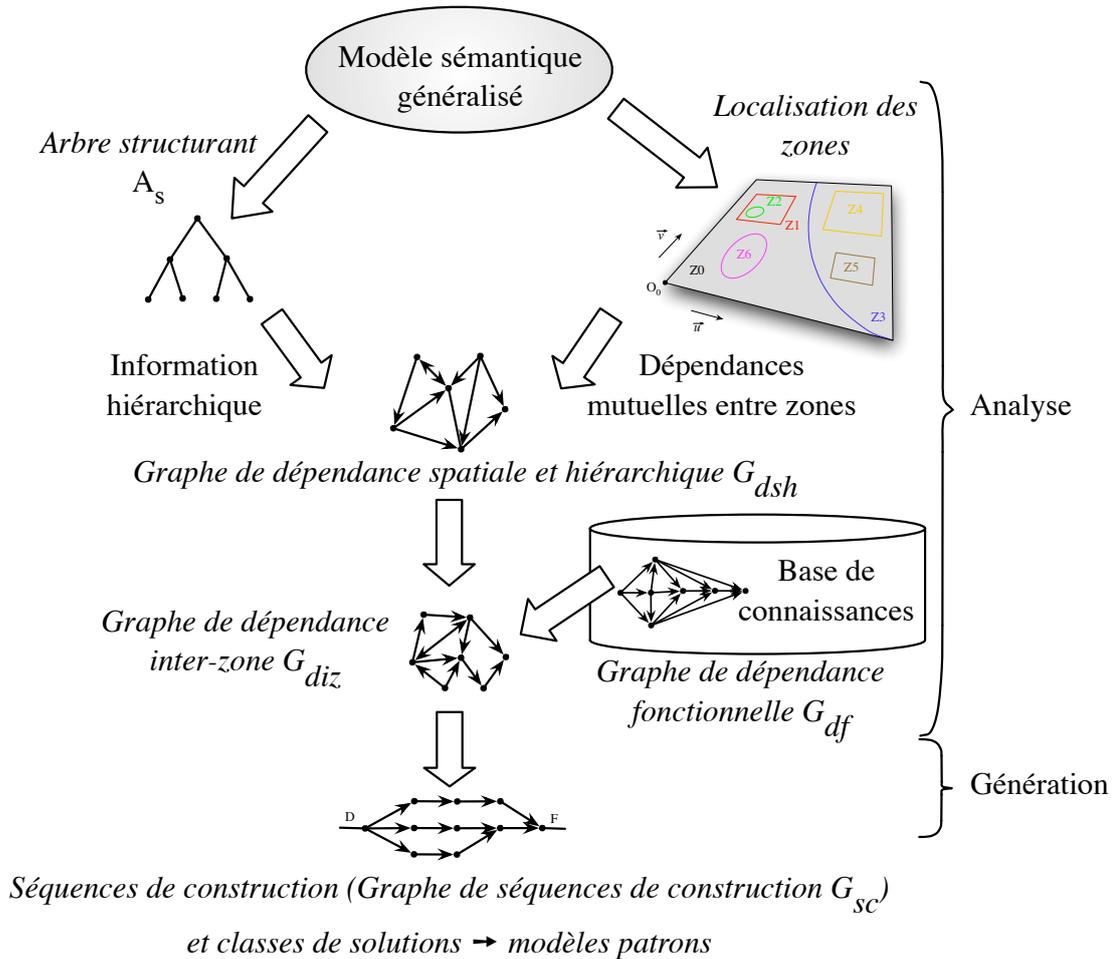


Figure 3.19. Graphes utilisés durant le processus de planification

L'ensemble des parties qui sont développées dans la suite de ce chapitre font intervenir des notions de la théorie des graphes. Le lecteur pourra se rapporter aux ouvrages [Berge, 1970, Gondran et Minoux, 1979, Labelle, 1981] pour avoir plus de détails sur ce domaine. Pour mieux comprendre les opérations réalisées, nous parlons de lien simple dans le cas d'un lien orienté entre deux nœuds, et de lien double lorsque cette orientation est réciproque. Voyons maintenant comment se réalise la construction de tous ces graphes en commençant par la phase d'analyse.

3.3.2.1 Analyse post-génération

La première étape de la phase d'analyse consiste à centraliser toutes les informations spatiales et hiérarchiques du modèle sémantique généralisé afin d'obtenir le graphe intermédiaire G_{dsh} . La construction de ce graphe se base sur l'arbre structurant A_s qui contient déjà l'information hiérarchique entre zones à travers l'architecture XML du modèle sémantique généralisé. Il faut alors lui ajouter les données de dépendances mutuelles entre zones et de le faire ainsi évoluer en G_{dsh} . La méthode utilisée pour cette opération est décrite dans le paragraphe suivant.

3.3.2.1.1 Évolution de \mathcal{A}_s en \mathcal{G}_{dsh}

La méthode que nous proposons pour faire évoluer \mathcal{A}_s en \mathcal{G}_{dsh} se fait en deux temps :

Étape 1 : prise en compte de la hiérarchie entre zones : parcourir l'arbre \mathcal{A}_s et ajouter à chaque nœud un lien simple orienté vers tous ses nœuds fils directs ou indirects. Cette opération correspond à considérer \mathcal{A}_s comme un graphe où toutes les arêtes sont orientées d'une zone mère vers ses zones filles et de calculer la fermeture transitive de ce graphe. Ce premier point traduit le fait simple de la hiérarchie des zones et la contrainte d'ordre de l'application des déformations de la zone mère avant celles de la zone fille. En effet, nous admettons qu'une zone mère a une influence sur toutes ses filles, mais qu'aucune de ses filles n'a d'influence sur elle. C'est pourquoi nous n'utilisons ici que des liens simples. De plus, cette représentation reflète bien la transitivité de l'influence dans la hiérarchie. En effet, si $Z1$ est mère de $Z2$ et que $Z2$ est mère de $Z3$, alors $Z1$ est «grand-» mère de $Z3$. $Z1$ exerce donc bien son influence sur $Z3$. Notons que par ce principe, la zone racine $Z0$ a un lien simple vers toutes les zones définies sur la surface. La figure 3.20 illustre cette première étape ;

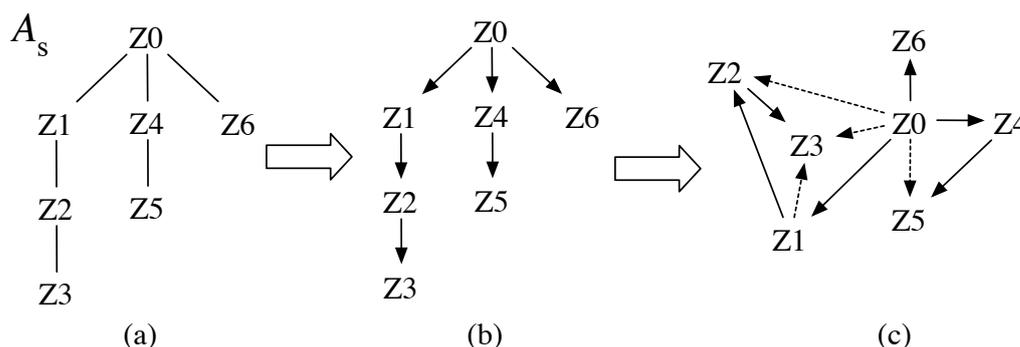


Figure 3.20. Étape 1 : prise en compte de la hiérarchie, (a) représente l'arbre \mathcal{A}_s , (b) est le graphe orienté selon la hiérarchie des zones, (c) est la fermeture transitive du graphe (b) où les traits en pointillés sont les liens ajoutés par l'opération

Étape 2 : prise en compte des chevauchements entre zones. Pour cela, il faut ajouter un lien à double orientation entre deux nœuds qui n'ont pas de relations hiérarchiques et qui correspondent à deux zones en dépendance mutuelle. Nous considérons ici qu'une zone est en dépendance mutuelle avec elle-même pour gérer le cas de plusieurs déformations sur une même zone. Ainsi chaque zone a un lien double qui la relie à elle-même. Le double lien traduit la dépendance mutuelle entre les zones.

Exemple : Soit un modèle sémantique généralisé répondant à la hiérarchie de l'arbre structurant de la figure 3.20.a et aux positions des zones schématisées sur la figure 3.21.a. L'étape 2 consiste à compléter le graphe issu de l'étape 1, c'est-à-dire le graphe de la figure 3.20.c en incluant des doubles liens entre les zones en chevauchement. Selon la figure 3.21.a, il

existe des recouvrements de zones entre : $Z1$ et $Z6$, $Z2$ et $Z6$, $Z3$ et $Z6$ ainsi que $Z4$ et $Z6$. Les chevauchements de zones ayant une relation hiérarchique comme $Z1$ et $Z3$ par exemple ne sont pas concernés. Ces dépendances mutuelles sont détectées par la technique précédemment citée et les doubles liens correspondants sont ajoutés au graphe de la figure 3.20.c, on obtient alors le graphe \mathcal{G}_{dsh} présenté sur la figure 3.21.b.

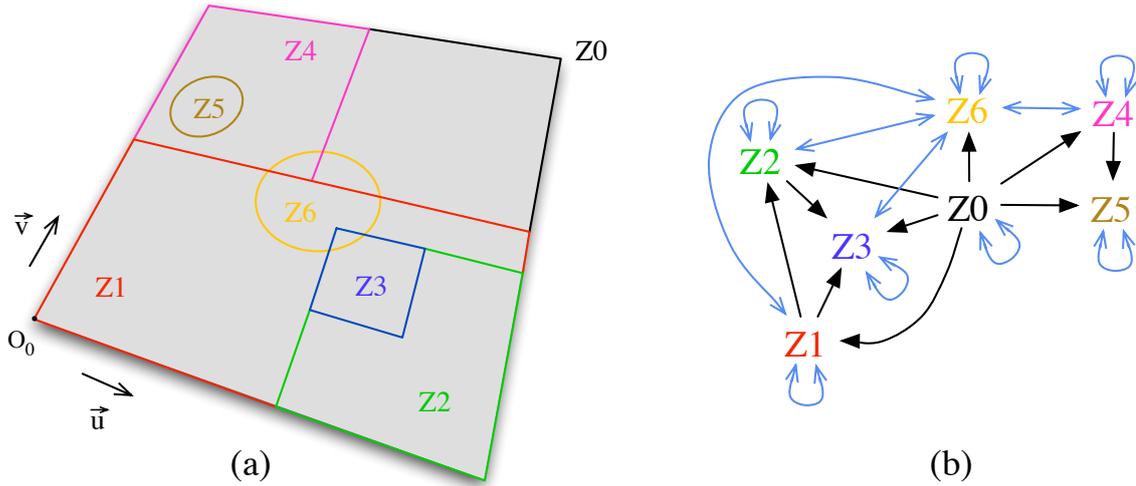


Figure 3.21. Ex : Plan paramétrique localisant les zones (a) et graphe \mathcal{G}_{dsh} résultat (b)

Voyons maintenant comment à partir de ce graphe \mathcal{G}_{dsh} combiné avec le graphe de dépendances fonctionnelles contenu dans la base de connaissances, nous obtenons le graphe de dépendances inter-zones.

3.3.2.1.2 Construction du graphe de dépendances inter-zones \mathcal{G}_{diz}

Le graphe de dépendances inter-zones doit pouvoir modéliser toutes les séquences de construction différentes possibles correspondant à un modèle sémantique généralisé. Il doit donc intégrer à la fois les contraintes de dépendances mutuelles et hiérarchiques entre zones (grâce à \mathcal{G}_{dsh}) et les contraintes d'ordonnancement des déformations (grâce à \mathcal{G}_{df}). Ce graphe est donc en quelque sorte une «fusion» des connaissances contenues \mathcal{G}_{dsh} et \mathcal{G}_{df} .

Sa construction commence alors par la définition de l'ensemble de ses nœuds. Un nœud de \mathcal{G}_{diz} correspond à un couple (*Zone* Z , *Déformation* \mathcal{D}) qui peut se lire : «appliquer la déformation \mathcal{D} sur la zone Z ». Ces couples sont déterminés à partir des informations contenues dans le modèle sémantique généralisé courant. Ainsi, si la zone 2 est associée à la déformation «BOMBE», alors \mathcal{G}_{diz} contiendra le nœud ($Z2, Bb$). Dans le cas où une zone est associée à plusieurs déformations, alors un couple est créé pour chaque déformation. Par exemple, si la zone 2 est bombée et bosselée, alors \mathcal{G}_{diz} contiendra les nœuds ($Z2, Bb$) et ($Z2, Bo$).

Il faut maintenant lier tous les nœuds de \mathcal{G}_{diz} . La méthode consiste à ajouter au graphe \mathcal{G}_{diz} les liens figurant dans \mathcal{G}_{df} qui existent aussi dans \mathcal{G}_{dsh} . Nous considérons ainsi que ce sont les contraintes d'ordonnancement des déformations qui sont prioritaires. Ce choix est réalisé car c'est l'ordre des déformations qui conditionne le résultat géométrique final. \mathcal{G}_{diz} ne doit pas contenir de lien d'un nœud vers lui-même car une opération (*Zone, Déformation*) ne peut apparaître qu'une seule fois dans une séquence de construction. Pour cela, en se basant sur les déformations lorsque l'on travaille sur \mathcal{G}_{df} et sur les zones lorsque l'on travaille sur \mathcal{G}_{dsh} , il faut appliquer les règles suivantes :

1. S'il n'y a pas le lien (simple ou double) concerné dans \mathcal{G}_{dsh} , c'est que les zones sont indépendantes, l'ordre d'application des déformations n'a pas d'importance, aucun lien n'est donc à ajouter dans \mathcal{G}_{diz} . Cette règle est prioritaire face à toutes les autres ;
2. Si le lien concerné dans \mathcal{G}_{df} (liant les déformations) est double, alors le lien à ajouter dans \mathcal{G}_{diz} est de même nature (simple ou double) et a la même orientation que celui concerné dans \mathcal{G}_{dsh} (liant les zones).

Ainsi, il n'y a pas de contrainte d'ordre sur les déformations concernées (lien double dans \mathcal{G}_{df}). Mais cela signifie aussi que celles-ci sont non commutatives. Il faut donc tenir compte du fait que l'on peut obtenir des résultats différents si les zones se chevauchent (lien double dans \mathcal{G}_{dsh}) en ajoutant un lien double dans \mathcal{G}_{diz} . Mais il faut aussi tenir compte que si une zone est mère d'une autre (lien simple dans \mathcal{G}_{dsh}), il faut appliquer les déformations de la zone mère avant celles de la zone fille, d'où l'ajout d'un lien simple orienté de la zone mère vers la zone fille dans \mathcal{G}_{diz} .

Exemple : Soit deux nœuds de \mathcal{G}_{diz} à lier : $(Z1, Cb)$ et $(Z2, Bb)$ ($Z1$ doit être courbée et $Z2$ doit être bombée). Les déformations Bb et Cb n'étant pas commutatives entre elles, si l'on applique Bb puis Cb sur une même région d'une surface, on peut ne pas obtenir le même résultat que si on applique Cb puis Bb . Il existe donc un lien double entre Bb et Cb dans \mathcal{G}_{df} . Nous sommes donc bien dans le cas d'application de la règle 2.

Cas 1 : Admettons que $Z1$ et $Z2$ se chevauchent sans avoir de relation hiérarchique entre-elles. Elles sont donc en dépendance mutuelle et il y a un lien double entre $Z1$ et $Z2$ dans \mathcal{G}_{dsh} . Dans ce cas, il faut que \mathcal{G}_{diz} tienne compte du fait qu'appliquer Cb sur $Z1$ puis Bb sur $Z2$ peut donner un résultat différent de l'application de Bb sur $Z2$ puis Cb sur $Z1$. La prise en compte de ces deux possibilités, ou deux ordres, passe par l'ajout d'un lien double entre les nœuds $(Z1, Cb)$ et $(Z2, Cr)$ dans \mathcal{G}_{diz} ;

Cas 2 : Supposons maintenant que $Z1$ est la zone mère de $Z2$. Cela se traduit par un lien simple de $Z1$ vers $Z2$ dans \mathcal{G}_{dsh} . Ce lien signifie que l'application des déformations sur $Z1$ sont prioritaires par rapport à celles à appliquer sur $Z2$. Sachant qu'il n'y a pas de contrainte de priorité entre Cb et Bb , seul l'ordre d'application de Cb sur $Z1$ puis Bb sur $Z2$ est à prendre en compte. Ainsi cela se traduit par un lien simple dans \mathcal{G}_{diz} de $(Z1, Cb)$ vers $(Z2, Bb)$;

3. Si le lien concerné dans \mathcal{G}_{df} (liant les déformations) est un lien simple, on ne tient pas compte du lien concerné dans \mathcal{G}_{dsh} et on ajoute à \mathcal{G}_{diz} un lien simple dont l'orientation est la même que celle du lien concerné dans \mathcal{G}_{df} .

Cette règle traduit la priorité des contraintes d'ordre des déformations sur celles concernant le hiérarchie.

Exemple : Soit deux nœuds de \mathcal{G}_{diz} à lier : $(Z1, Bo)$ et $(Z2, Bb)$ ($Z1$ doit être bosselée et $Z2$ doit être bombée). Nous savons que l'opération consistant à bosseler (Bo) est une opération que nous qualifions de finition qui doit être réalisée en dernier. Elle doit donc en particulier être appliquée après une déformation Bb . Cela est traduit dans \mathcal{G}_{df} par un lien simple de Bb vers Bo . Nous sommes donc bien dans le cas d'application de la règle 3. Que $Z1$ soit la zone mère de $Z2$ (lien simple dans \mathcal{G}_{df}) ou qu'elles soient en dépendances mutuelles (lien double dans \mathcal{G}_{df}), la déformation Bo reste une déformation de finition et doit toujours être appliquée après l'opération de Bb . Un lien simple est donc ajouté à \mathcal{G}_{diz} de $(Z2, Bb)$ vers $(Z1, Bo)$;

4. S'il n'existe pas de lien dans \mathcal{G}_{df} entre les déformations concernées, c'est que l'on est forcément⁹ dans le cas où les déformations en question sont identiques et considérées comme commutatives entre elles, selon la remarque 3.2.1. Ainsi, il n'y a pas de contrainte sur l'ordre à appliquer entre ces deux déformations puisque l'on sait que le résultat sera le même du fait de leur commutativité. Il faut donc d'ajouter un lien simple dans \mathcal{G}_{diz} dont l'orientation est choisie arbitrairement.

Le graphe \mathcal{G}_{diz} ainsi obtenu modélise toutes les séquences possibles de construction où chaque arc correspond à une relation d'antériorité possible.

Exemple : Pour illustrer la construction du graphe de dépendances inter-zones, nous choisissons un exemple simple dont l'arbre structurant \mathcal{A}_s est représenté sur la figure 3.22.a et le placement des zones dans le plan paramétrique est schématisé sur la figure 3.22.b.

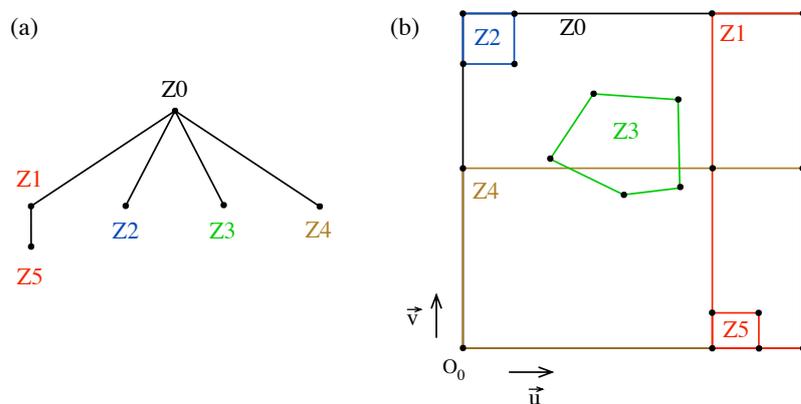


Figure 3.22. Ex 2 : Arbre structurant (a), plan paramétrique avec frontières des zones (b)

⁹Si les déformations considérées sont différentes, on est en présence d'un graphe mal formé selon la remarque 3.2.2.

Il est alors possible d'exprimer les informations données dans ces deux schémas par les ensembles suivants, où \mathcal{Z} est l'ensemble des zones filles de la zone courante \mathcal{Z}_i et \mathcal{D} est l'ensemble des déformations attachées à \mathcal{Z}_i :

$$\begin{aligned} \mathcal{Z}_0 &: \mathcal{Z} = \{ \mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3, \mathcal{Z}_4, \mathcal{Z}_5 \}, & \mathcal{D} &= \{ Bo \} \\ \mathcal{Z}_1 &: \mathcal{Z} = \{ \mathcal{Z}_5 \}, & \mathcal{D} &= \{ Bb \} \\ \mathcal{Z}_2 &: \mathcal{Z} = \emptyset, & \mathcal{D} &= \{ Cr \} \\ \mathcal{Z}_3 &: \mathcal{Z} = \emptyset, & \mathcal{D} &= \{ Tr \} \\ \mathcal{Z}_4 &: \mathcal{Z} = \emptyset, & \mathcal{D} &= \{ Bb \} \\ \mathcal{Z}_5 &: \mathcal{Z} = \emptyset, & \mathcal{D} &= \{ Cr \} \end{aligned}$$

En utilisant les informations de recouvrement des zones de la figure 3.22 et en se basant sur l'arbre structurant, il est possible de construire le graphe de dépendances spatiales et hiérarchiques \mathcal{G}_{dsh} . Pour cela, il faut transformer \mathcal{A}_s en graphe orienté (Fig. 3.23.a), puis calculer sa fermeture transitive (Fig. 3.23.b). Enfin il faut intégrer les chevauchements de zones. Ici cette action permet d'enrichir le graphe de la figure 3.23.b de neuf liens doubles : entre \mathcal{Z}_1 et \mathcal{Z}_4 , \mathcal{Z}_5 et \mathcal{Z}_4 , entre \mathcal{Z}_3 et \mathcal{Z}_4 et enfin les liens doubles des zones vers elles-mêmes. Le graphe ainsi obtenu est \mathcal{G}_{dsh} et correspond à la figure 3.23.c.

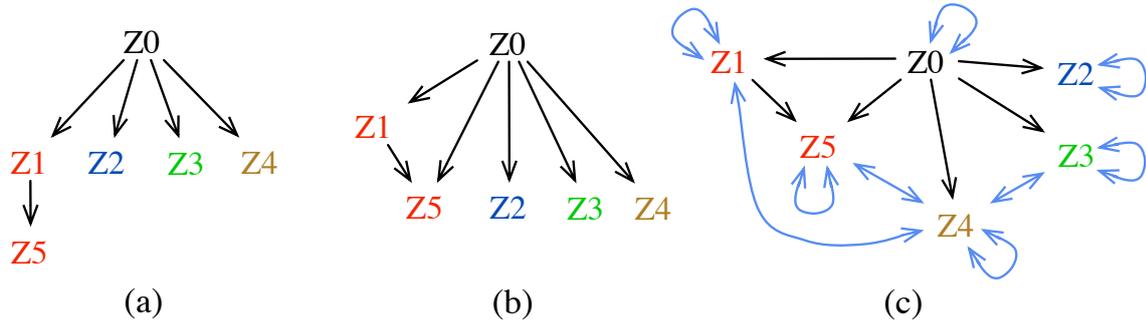


Figure 3.23. Ex 2 : de \mathcal{A}_s vers \mathcal{G}_{dsh}

Le graphe de dépendances fonctionnelles issu de la base de connaissances est partiellement représenté sur la figure 3.24. Il répond à la légende correspondant aux déformations suivantes : Cr : Creux, Bb : Bombé, Tr : Troué, Bo : Bosselé.

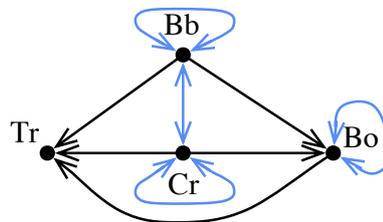


Figure 3.24. Ex 2 : extrait de \mathcal{G}_{df} contenu dans la base de connaissances

Fort de \mathcal{G}_{dsh} et de \mathcal{G}_{df} , il est possible de construire \mathcal{G}_{diz} . Selon le modèle sémantique généralisé, la surface est constituée de six zones, chaque zone ayant une seule déformation

qui lui est rattachée. Ainsi le graphe de dépendances inter-zones est construit à partir de six nœuds qui sont : (Z_0,Bo) ; (Z_1,Bb) ; (Z_2,Cr) ; (Z_3,Tr) ; (Z_4,Bb) et $(Z_5;Cr)$. Il faut maintenant ajouter les liens à partir des règles de construction de \mathcal{G}_{diz} :

Règle 1 : Il existe dans \mathcal{G}_{dsh} des liens entre :

Z_0 et Z_1, Z_2, Z_3, Z_4, Z_5 ;

Z_1 et Z_4, Z_5 ;

Z_3 et Z_4 ;

Z_4 et Z_5 .

Il ne faut donc traiter que les liens concernant ces couples dans les autres règles ;

Règle 2 : Il existe dans \mathcal{G}_{df} des liens doubles entre Bb et elle-même, et entre Bb et Cr , donc on ajoute à \mathcal{G}_{diz} les liens liant les zones Z_1, Z_2, Z_4 et Z_5 entre elles dans \mathcal{G}_{dsh} soit :

Lien simple de (Z_1,Bb) vers $(Z_5;Cr)$; Liens doubles entre (Z_1,Bb) et (Z_4,Bb) et entre (Z_4,Bb) et $(Z_5;Cr)$;

Règle 3 : Il existe dans \mathcal{G}_{df} un lien simple de Cr vers Bo , on ajoute donc un lien simple de (Z_2,Cr) vers (Z_0,Bo) et de $(Z_5;Cr)$ vers (Z_0,Bo) . On trouve dans \mathcal{G}_{df} un lien simple de Bb vers Bo on ajoute donc à \mathcal{G}_{diz} un lien simple de (Z_1,Bb) vers (Z_0,Bo) et de (Z_4,Bb) vers (Z_0,Bo) . Il existe dans \mathcal{G}_{df} un lien simple de Bb vers Tr , on ajoute donc un lien simple de (Z_4,Bb) vers (Z_3,Tr) . Enfin dans \mathcal{G}_{df} , on trouve un lien simple de Bo vers Tr , on ajoute donc dans \mathcal{G}_{diz} un lien simple de (Z_0,Bo) vers (Z_3,Tr) .

Tous les liens de \mathcal{G}_{dsh} ont été traités, la construction de \mathcal{G}_{diz} est donc terminée, celui-ci est présenté sur la figure 3.25.

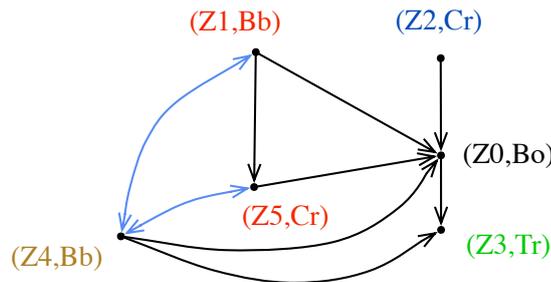


Figure 3.25. Exemple : Graphe de dépendances inter-zones \mathcal{G}_{diz}

Remarque 3.3.1 – Si toutes les zones sont en dépendance mutuelle et qu’il n’existe pas de contrainte d’ordre dans le graphe de dépendances fonctionnelles entre les déformations concernées, le graphe de dépendances inter-zones est complet. Dans ce cas, il y a $N(N - 1)$ dépendances mutuelles (nombre d’arêtes de \mathcal{G}_{diz}) où N est le nombre de zones, et il y a $N!$ classes de solutions possibles. Néanmoins, s’il y a indépendance entre toutes les zones il n’existe qu’une seule classe de solutions. En pratique, il est très rare d’avoir toutes les zones qui se chevauchent en un même région ce qui réduit considérablement le nombre de classes

de solutions. Il existe cependant des configurations de type «chaîne» qui peuvent très vite faire augmenter le nombre de classes solutions en propageant les dépendances de zones en zones.

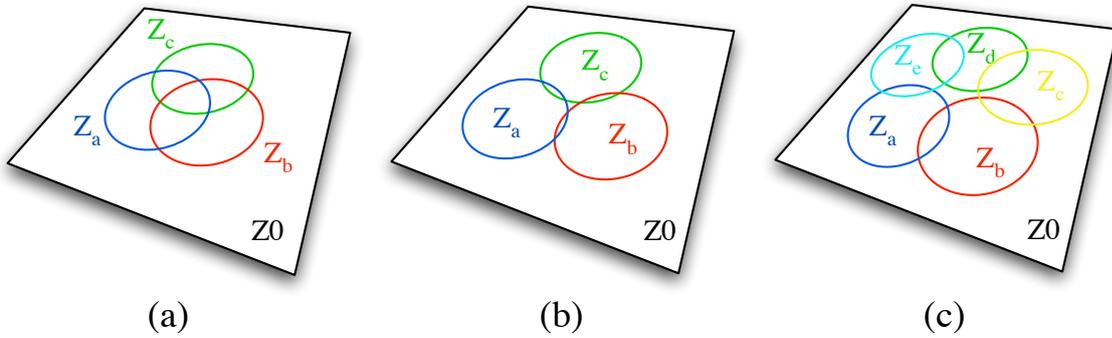


Figure 3.26. Exemples de configurations de type «Chaîne fermée»

La figure 3.26 présente des exemples de configurations en «chaîne fermée». Le placement des zones sur les figures 3.26.a et 3.26.b est différent. Sur la première il y a une région où les trois zones se chevauchent alors que sur la seconde les zones ne se chevauchent que deux à deux. Malgré cette différence de taille, le phénomène de chaînage entraîne, sur ces deux exemples simples, un nombre de classes solutions identique qui s'élève à six, soit $3!$ pour les trois zones impliquées associée chacune à une déformation de forte dépendance dans \mathcal{G}_{df} . La figure 3.26.c, quant à elle, présente une configuration faisant intervenir cinq zones en chaîne fermée.

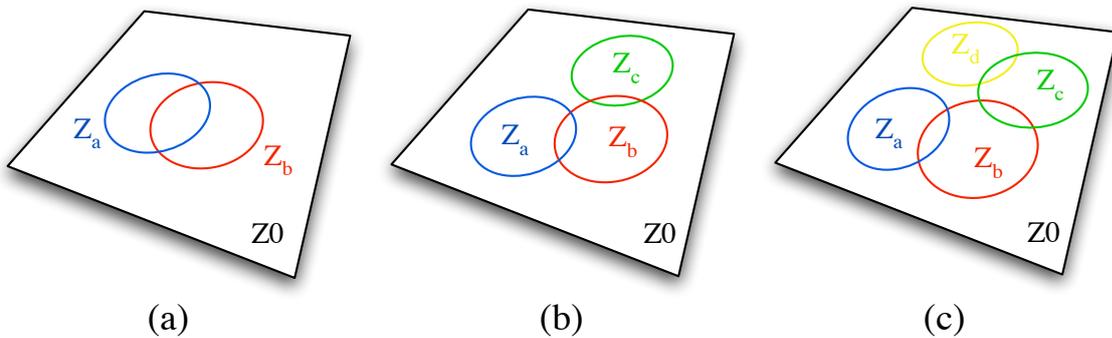


Figure 3.27. Exemples de configurations de type «Chaîne ouverte»

Le cas des configurations de zones en «chaînes ouvertes» présentées sur la figure 3.27 voit aussi son nombre de classes solutions augmenter très vite avec le nombre de zones impliquées. Sa croissance est cependant simple à évaluer puisqu'à chaque ajout d'une zone dans la chaîne, le nombre de classes solutions est alors multiplié par deux dans le pire des cas. En effet, ajouter une zone à la chaîne, dans le pire des cas, équivaut à créer un lien double dans \mathcal{G}_{diz} entre la nouvelle zone et une des extrémités de la chaîne. Ce cas se traduit dans la génération des modèles patrons (Cf. 3.3.2.2) par le dédoublement des classes

de solutions. Ainsi dans le cas des configurations de zones en chaînes ouvertes, le nombre de classes solutions est de 2^{N-1} où N est le nombre de zones impliquées dans la chaîne ouverte.

On s'aperçoit donc assez vite que selon la position des zones et le type de déformation utilisé, le nombre de classes de solutions peut très rapidement devenir difficile à gérer. C'est pourquoi il est essentiel dans une approche telle que l'approche déclarative de penser aux stratégies de parcours de l'espace des solutions. Cette partie n'a pas été traitée dans le cadre de cette thèse, mais des pistes de recherche et des orientations de travail sont proposées dans les perspectives à la fin de ce document.

C'est à partir du graphe de dépendances inter-zones que les séquences de construction et donc les modèles patrons, peuvent être déterminés. La partie suivante décrit la méthode que nous mettons en œuvre pour réaliser cette tâche.

3.3.2.2 Génération des modèles patrons

La génération des modèles patrons consiste à trouver toutes les façons possibles d'appliquer les déformations, tout en respectant les contraintes d'ordre fixées dans le graphe de dépendances inter-zones \mathcal{G}_{diz} . Chaque ordre est représenté par une séquence de construction. Plusieurs séquences possibles peuvent être équivalentes et ainsi appartenir à une même classe de solutions. L'objectif est donc de trouver toutes ces séquences de construction possibles tout en les regroupant en classes d'équivalence. Pour cela, nous faisons intervenir la notion d'intervalle à l'intérieur d'une séquence de construction :

Définition 3.3.5 – On appelle **intervalle** d'une séquence de construction \mathcal{S}_i , tout couple (ZD_1, ZD_2) , où ZD_1 et ZD_2 sont des couples (Zone, Déformation) appartenant à \mathcal{S}_i tels que ZD_1 est placé avant ZD_2 dans la séquence \mathcal{S}_i . ZD_1 et ZD_2 peuvent ne pas être consécutifs. La taille d'un intervalle se calcule par le nombre d'éléments qu'il contient (en comptant ses extrémités) moins 1. Un tel intervalle est considéré comme vide si ZD_1 est placé après ZD_2 dans la séquence \mathcal{S}_i ou si $ZD_1 = ZD_2$. Par opposition, si l'on considère qu'une séquence commence toujours par le symbole D et se termine toujours par le symbole F , alors le couple (D, F) représente la séquence complète et sa taille est égale à celle de la séquence.

La méthode que nous proposons pour construire l'ensemble \mathcal{S} de toutes les séquences de construction possibles associées à \mathcal{G}_{diz} , consiste à placer un à un les nœuds de \mathcal{G}_{diz} à toutes les positions qu'ils peuvent tenir dans \mathcal{S} . Ce placement se fait en quatre étapes :

1. Initialiser l'ensemble des séquences de construction. Cette étape consiste à considérer \mathcal{S} avec une seule séquence de construction associée à une première classe. Cette séquence est constituée d'un nœud de \mathcal{G}_{diz} tiré au hasard. Ce nœud est considéré comme placé.

Par cette action, on crée la première séquence qui va servir de point de départ et de référent pour l'ajout des autres opérations ZD ;

2. Choisir un nœud ZD de \mathcal{G}_{diz} à placer. Le nœud en question correspond à une opération (*Zone, Déformation*) qui peut être appliquée avant ou après les opérations déjà prévues dans les séquences. Il faut donc choisir un nœud de \mathcal{G}_{diz} qui est en relation (ou voisin) avec des nœuds déjà traités, ceci afin de pouvoir le placer relativement aux opérations déjà prévues ;
3. Trouver où et comment placer ZD en tenant compte des contraintes d'ordonnement (liens simples) contenus dans \mathcal{G}_{diz} . Pour cela, il faut trouver toutes les opérations devant être obligatoirement appliquées avant ZD (lien simple dirigé vers ZD). On obtient ainsi l'ensemble B_f (pour borne de début). Mais il faut aussi trouver toutes les opérations devant être obligatoirement réalisées après ZD (lien simple partant de ZD), on obtient ainsi l'ensemble B_d (pour borne de fin). L'opération ZD doit donc être appliquée entre une des opérations de B_d et une des opérations de B_f . Dans le cas où $B_d = \emptyset$, on considère que $B_d = D$, c'est-à-dire que ZD peut être appliquée dès le début d'une séquence. Dans le cas où $B_f = \emptyset$, on considère que $B_f = F$, c'est-à-dire que ZD peut être appliquée jusqu'à la fin d'une séquence. Le produit cartésien noté \times de ces ensembles nous donne donc tous les intervalles possibles dans lesquels ZD peut être placé. On note I_k ces intervalles possibles. Ainsi si, par exemple, ZD doit être inséré après $ZD1$, après $ZD3$ et avant $ZD4$, on est dans le cas précis où $B_d = \{ZD1, ZD3\}$ et $B_f = \{ZD4\}$. Le produit cartésien de B_d et B_f est $B_d \times B_f = \{\{ZD1, ZD4\}\{ZD3, ZD4\}\}$. Il existe donc deux intervalles possibles $I_1 = (ZD1, ZD4)$ et $I_2 = (ZD3, ZD4)$;
4. Trouver les nœuds pivots de l'intervalle, s'il en existe. Ces nœuds pivots reflètent le changement de classe d'équivalence et permettent donc de dire que si ZD est appliquée avant l'opération du nœud pivot, le résultat obtenu peut être différent de celui obtenu en appliquant ZD après l'opération du nœud pivot. Cette notion est directement liée à celle de non commutativité des déformations qui est stockée sous la forme des liens doubles dans \mathcal{G}_{diz} . Les nœuds pivots sont donc tous les nœuds reliés à ZD dans \mathcal{G}_c par des liens doubles ;
5. Placer ZD dans toutes les séquences de \mathcal{S} (ne contenant pas ZD) à l'intérieur de l'intervalle I_k minimum tout en tenant compte des nœuds pivots. Toutes les séquences de \mathcal{S} étant différentes, tous les intervalles I_k sont aussi différents. Nous choisissons l'intervalle minimum que l'on note I afin de respecter toutes les contraintes d'ordre à la fois. En effet, si ZD doit être inséré après $ZD1$ et après $ZD3$ alors que $ZD3$ est après $ZD1$ dans la séquence, il faut alors le placer après $ZD3$ pour respecter les deux contraintes en même temps. Cette démarche est la même pour la borne de fin de l'intervalle ce qui minimise sa taille.
Il faut maintenant placer ZD entre chaque nœud de I . Chaque insertion entre deux nœuds de I implique la création d'une nouvelle séquence dans \mathcal{S} . En effet, il faut prévoir tous les cas de figures possibles afin de proposer toutes les solutions possibles

aux prochains nœuds à placer.

Ainsi si $I = (ZD_2, ZD_4)$ et que $S_i = \{ZD_1, ZD_2, ZD_3, ZD_4, ZD_5\}$, alors les séquences $\{ZD_1, ZD_2, \mathbf{ZD}, ZD_3, ZD_4, ZD_5\}$ et $\{ZD_1, ZD_2, ZD_3, \mathbf{ZD}, ZD_4, ZD_5\}$ sont créées dans \mathcal{S} .

Tant que les insertions sont faites avant le premier nœud pivot de I , la séquence créée est associée à la même classe d'équivalence que S_i . Dès que l'on passe un nœud pivot, les séquences créées jusqu'au prochain nœud pivot sont associées à une nouvelle classe d'équivalence. Cette nouvelle classe d'équivalence sera la même lors du passage du même pivot pour toutes les séquences S_i appartenant à la même classe d'équivalence lors du traitement du même nœud ZD .

Si par exemple $S_i = \{ZD_1, ZD_2, ZD_3, ZD_4, ZD_5\}$ appartient à la classe 0, que $I = (ZD_1, F)$ et que ZD_3 et ZD_5 sont des nœuds pivots de I , alors les séquences suivantes seront créées sans \mathcal{S} :

$\{ZD_1, \mathbf{ZD}, ZD_2, ZD_3, ZD_4, ZD_5\}$ associée à la classe 0 ;

$\{ZD_1, ZD_2, \mathbf{ZD}, ZD_3, ZD_4, ZD_5\}$ associée à la classe 0 ;

On va dépasser le nœud pivot ZD_3 , on crée la classe 1 ;

$\{ZD_1, ZD_2, ZD_3, \mathbf{ZD}, ZD_4, ZD_5\}$ associée à la classe 1 ;

$\{ZD_1, ZD_2, ZD_3, ZD_4, \mathbf{ZD}, ZD_5\}$ associée à la classe 1 ;

On va dépasser le nœud pivot ZD_5 , on crée la classe 2 ;

$\{ZD_1, ZD_2, ZD_3, ZD_4, ZD_5, \mathbf{ZD}\}$ associée à la classe 2 ;

Lorsqu'une séquence est traitée, on la supprime pour ne garder que les nouvelles séquences intégrant le nœuds ZD , on passe à une nouvelle séquence et on applique à nouveau l'étape 5 ;

6. Lorsque toutes les séquences de \mathcal{S} contiennent ZD , ZD est considéré comme placé et on passe à un autre nœud de \mathcal{G}_{diz} en repartant de l'étape 2.

Lorsque tous les nœuds de \mathcal{G}_{diz} sont traités, \mathcal{S} contient toutes les séquences possibles répondant à \mathcal{G}_{diz} , regroupées en classes d'équivalence. Il faut alors choisir une séquence représentant chaque classe pour obtenir tous les modèles patrons associés à \mathcal{G}_{diz} .

Exemple : Reprenons le graphe \mathcal{G}_{diz} de l'exemple précédent présenté à la figure 3.25.

Étape 1 : On tire au hasard un nœud de \mathcal{G}_{diz} , par exemple $(Z5, Cr)$, on crée une première séquence \mathcal{S}_1 associée à la classe 0 que l'on remplit avec ce nœud. Ainsi, on a la première séquence telle que :

$0 \Rightarrow (Z5, Cr)$;

On peut alors représenter le graphe des nœuds traités en marquant les nœuds sur \mathcal{G}_{diz} comme le montre la figure 3.28.a ou en reconstruisant ce graphe au fur et à mesure que les nœuds sont traités (Fig 3.28.b). Cette dernière représentation est nommée \mathcal{G}_c pour graphe de construction et permet de voir au fur et à mesure les nouvelles contraintes d'ordre à traiter.

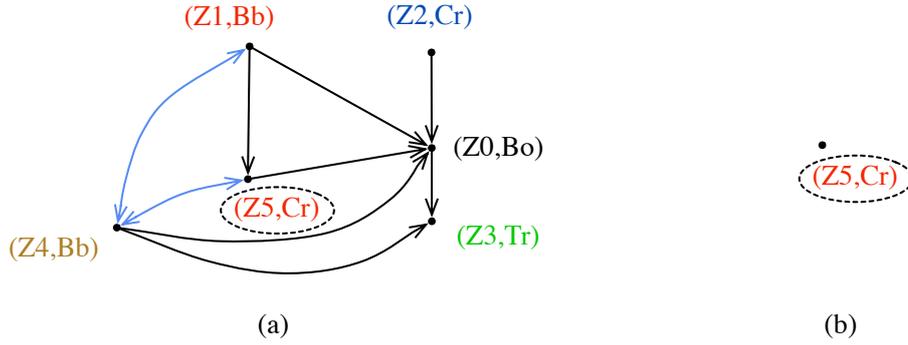


Figure 3.28. Construction des modèles patrons, étape d'initialisation

Étape 2 : Selon la figure 3.28, le nœud traité $(Z5,Cr)$ a 3 voisins qui sont $(Z4,Bb)$, $(Z0,Bo)$ et $(Z1,Bb)$. Une de ces opérations est arbitrairement choisie, par exemple $Z0,Bo$. La figure 3.29 illustre ce choix et souligne le nœud concerné.

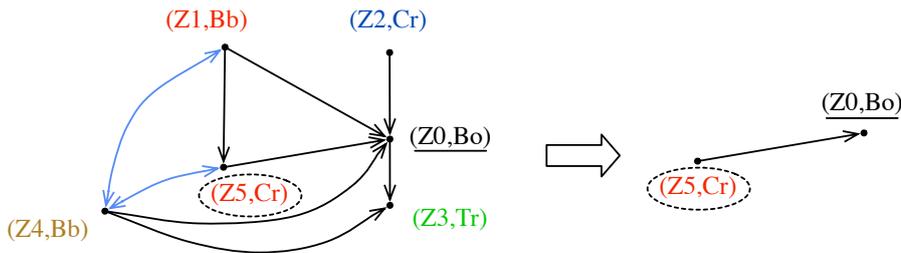


Figure 3.29. Construction des modèles patrons, choix d'un nœud à traiter

Étape 3 : Le but est de placer $(Z0,Bo)$, or selon la figure 3.29.b, il n'existe qu'un lien simple de $(Z5,Cr)$ vers $(Z0,Bo)$, ce qui veut dire que la déformation Cr appliquée sur $Z5$ doit être réalisée avant la déformation Bo appliquée sur $Z0$. Ainsi le seul intervalle possible où peut être inséré $(Z0,Bo)$ est l'intervalle $I_1 = ((Z5,Cr), F)$;

Étape 4 : Il n'y a pas de lien double dans \mathcal{G}_s , il n'y a donc pas de nœud pivot et la classe reste donc inchangée ;

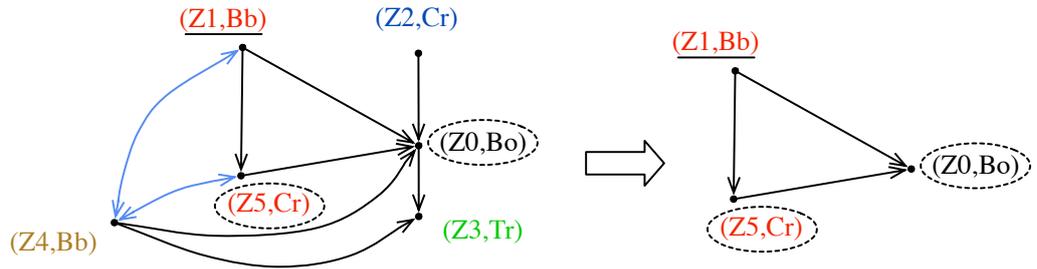
Étape 5 : Il faut ajouter à la première séquence le nœud $(Z0,Bo)$ dans l'intervalle I_1 , on obtient :

$$0 \Rightarrow (Z5,Cr) (Z0,Bo) ;$$

Étape 6 : On marque $(Z0,Bo)$ comme traité.

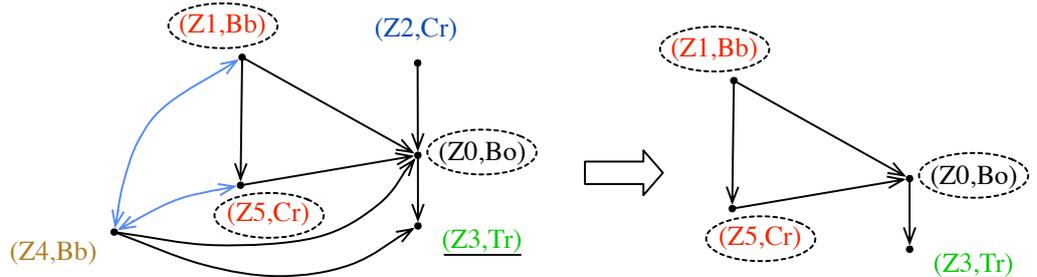
Il faut maintenant réitérer toutes ces étapes :

- On commence par choisir un nouveau nœud, prenons $(Z1,Bb)$ puisqu'il est voisin de $(Z0,Bo)$ et $(Z5,Cr)$.



Selon \mathcal{G}_s , il doit être placé avant $(Z5, Cr)$ et avant $(Z0, Bo)$. Il existe donc deux intervalles possibles : $I_1 = ((Z5, Cr), F)$ et $I_2 = ((Z0, Bo), F)$. Aucun lien double n'existe dans \mathcal{G}_s . Il n'y a donc pas de nœud pivot, la classe reste donc inchangée. Il n'existe qu'une séquence à traiter et sur cette séquence, I_2 a sa taille inférieure à celle de I_1 . On place **(Z1,Bb)** dans I_2 et on obtient :

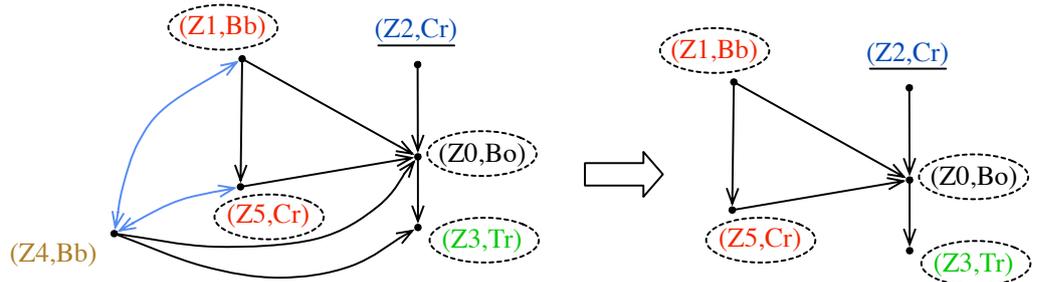
- 0 => **(Z1,Bb)** (Z5, Cr) (Z0, Bo) ;
- On choisit **(Z3,Tr)** car il est voisin de $(Z0, Bo)$.



Il doit être placé après $(Z0, Bo)$. Aucun nœud pivot n'est présent, la classe reste inchangée :

0 => (Z1, Bb) (Z5, Cr) (Z0, Bo) **(Z3, Tr)** ;

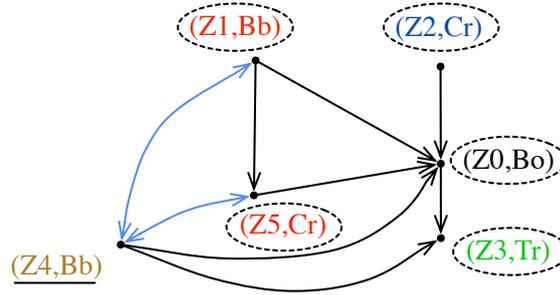
- On choisit **(Z2,Cr)** car il est voisin de $(Z0, Bo)$.



Il doit être placé avant $(Z0, Bo)$. Aucun nœud pivot n'est présent, la classe reste inchangée :

- 0 => **(Z2,Cr)** (Z1, Bb) (Z5, Cr) (Z0, Bo) (Z3, Tr) ; → séquence A (séqu. A)
- 0 => (Z1, Bb) **(Z2,Cr)** (Z5, Cr) (Z0, Bo) (Z3, Tr) ; → séquence B (séqu. B)
- 0 => (Z1, Bb) (Z5, Cr) **(Z2,Cr)** (Z0, Bo) (Z3, Tr) ; → séquence C (séqu. C)

- On choisit **(Z4,Bb)** car il est voisin de $(Z0, Bo)$, $(Z5, Cr)$, $(Z1, Bb)$ et $(Z3, Tr)$.



Il doit être placé avant (Z0,Bo) avec comme nœuds pivots (Z5,Cr) et (Z1,Bb) puisqu'il est lié à ces deux nœuds par les liens doubles dans \mathcal{G}_s . Il existe donc trois classes de solutions : avant (Z1,Bb), entre (Z1,Bb) et (Z5,Cr), entre (Z5,Cr) et (Z0,Bo) pour chacune des trois séquences existantes :

0 => (Z4,Bb) (Z2,Cr) (Z1,Bb) (Z5,Cr) (Z0,Bo) (Z3,Tr) ; 0 => (Z2,Cr) (Z4,Bb) (Z1,Bb) (Z5,Cr) (Z0,Bo) (Z3,Tr) ; On va dépasser le nœud pivot (Z1,Bb), on crée la classe 1 ; 1 => (Z2,Cr) (Z1,Bb) (Z4,Bb) (Z5,Cr) (Z0,Bo) (Z3,Tr) ; On va dépasser le nœud pivot (Z5,Cr), on crée la classe 2 ; 2 => (Z2,Cr) (Z1,Bb) (Z5,Cr) (Z4,Bb) (Z0,Bo) (Z3,Tr) ;	}	Issu de séq. A
0 => (Z4,Bb) (Z1,Bb) (Z2,Cr) (Z5,Cr) (Z0,Bo) (Z3,Tr) ; On va dépasser le nœud pivot (Z1,Bb), la séquence B appartient à la même classe d'équivalence que la séquence A, la nouvelle classe a déjà été créée, c'est la classe 1 ; 1 => (Z1,Bb) (Z4,Bb) (Z2,Cr) (Z5,Cr) (Z0,Bo) (Z3,Tr) ; 1 => (Z1,Bb) (Z2,Cr) (Z4,Bb) (Z5,Cr) (Z0,Bo) (Z3,Tr) ; On va dépasser le nœud pivot (Z5,Cr), la séquence B appartient à la même classe d'équivalence que la séquence A, la nouvelle classe a déjà été créée, c'est la classe 2 ; 2 => (Z1,Bb) (Z2,Cr) (Z5,Cr) (Z4,Bb) (Z0,Bo) (Z3,Tr) ;	}	Issu de séq. B
0 => (Z4,Bb) (Z1,Bb) (Z5,Cr) (Z2,Cr) (Z0,Bo) (Z3,Tr) ; On va dépasser le nœud pivot (Z1,Bb), la séquence C appartient à la même classe d'équivalence que la séquence A, la nouvelle classe a déjà été créée, c'est la classe 1 ; 1 => (Z1,Bb) (Z4,Bb) (Z5,Cr) (Z2,Cr) (Z0,Bo) (Z3,Tr) ; On va dépasser le nœud pivot (Z5,Cr), la séquence C appartient à la même classe d'équivalence que la séquence A, la nouvelle classe a déjà été créée, c'est la classe 2 ; 2 => (Z1,Bb) (Z5,Cr) (Z4,Bb) (Z2,Cr) (Z0,Bo) (Z3,Tr) ; 2 => (Z1,Bb) (Z5,Cr) (Z2,Cr) (Z4,Bb) (Z0,Bo) (Z3,Tr) ;	}	Issu de séq. C

Il existe donc trois classes d'équivalence pour cet exemple qui peuvent être représentées par trois modèles patrons :

- 0 => (Z4,Bb) (Z2,Cr) (Z1,Bb) (Z5,Cr) (Z0,Bo) (Z3,Tr) ;
- 1 => (Z1,Bb) (Z4,Bb) (Z5,Cr) (Z2,Cr) (Z0,Bo) (Z3,Tr) ;
- 2 => (Z1,Bb) (Z5,Cr) (Z4,Bb) (Z2,Cr) (Z0,Bo) (Z3,Tr) ;

En ne tenant compte que des modèles patrons choisis, il est possible de représenter les séquences de construction selon le graphe de la figure 3.30 que l'on nomme aussi graphe de séquences de construction \mathcal{G}_{sc} .

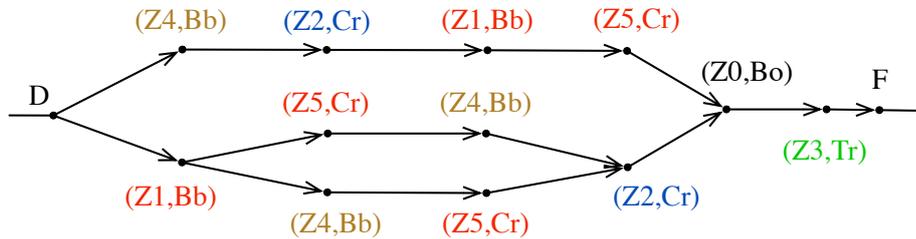


Figure 3.30. Exemple : Graphe de séquences de construction \mathcal{G}_{sc}

Dans ce graphe tout chemin possible partant de D et allant à F est une séquence possible répondant à \mathcal{G}_{diz} . Cette représentation ne permet cependant pas de visualiser le regroupement des séquences en classes d'équivalence.

3.4 Conclusion

L'objectif premier de l'approche déclarative de la modélisation de surfaces est de pouvoir engendrer une ou plusieurs surfaces par la seule donnée d'un ensemble de propriétés et de caractéristiques. Les moyens fournis à l'utilisateur pour exprimer ces propriétés et caractéristiques peuvent être de plusieurs ordres. C'est pourquoi nous introduisons la notion de modules d'interface. Ceux-ci ont pour but de traduire une description d'un type particulier dans un langage structuré et exploitable par la suite des traitements. Cette traduction s'exprime sous la forme d'un modèle appelé modèle sémantique. Celui-ci rassemble l'ensemble des informations, apportées par l'utilisateur, organisées en utilisant le langage XML. Généralement, les données fournies dans la description ne sont pas assez riches pour permettre de produire un modèle sémantique intégrant toutes les informations nécessaires à la production d'une surface solution. Ainsi nous introduisons un moteur de complétion sémantique qui, en utilisant une base de connaissances, complète les portions nécessaires du modèle sémantique qui lui est donné en entrée. Ce traitement passe par une phase de complétion symbolique puis d'une phase de complétion numérique. Dans cette dernière, seul le positionnement absolu des zones est traité, le positionnement relatif restant une perspec-

tive d'évolution n'entrant pas dans les délais de cette thèse. La complétion sémantique se termine alors par la génération de plusieurs modèles sémantiques généralisés qui sont ensuite exploités par le processus suivant que nous nommons «planification». Celui-ci a pour tâche de générer tous les modèles patrons associés à un modèle sémantique généralisé donné. Chaque modèle patron est le représentant d'une classe particulière de solutions définie par sa séquence de déformations à appliquer aux zones de la surface.

À ce stade de la méthode, toutes les données sont connues pour pouvoir commencer à obtenir des modèles géométriques. Le chapitre suivant expose les techniques que nous avons choisies pour y parvenir.

CONSTRUCTION DES MODÈLES GÉOMÉTRIQUES

4.1 Introduction

L'utilisateur a donné sa description et le système de modélisation déclarative de surfaces a proposé un ensemble de modèles patrons potentiellement solutions. Chacun de ces modèles renferme la sémantique donnée par la description de l'utilisateur associée à celle déduite par le système ainsi qu'un ordre de déformations à appliquer, ce que l'on appelle aussi le plan de construction. Dans ce chapitre, nous montrons comment à partir d'un de ces modèles patrons, nous construisons un ou plusieurs modèles géométriques solutions. Dans la première partie de ce chapitre, nous commençons par introduire les méthodes existantes de manipulation et de déformation de courbes et de surfaces à pôles. Nous poursuivons par la présentation de notre système de résolution/construction en détaillant d'abord notre système de déformation par zone. Celui-ci permet d'appliquer, selon le plan de construction, les contraintes dites «lâches» souhaitées par l'utilisateur. Nous développons ensuite notre approche de satisfaction de contraintes de passages, ou contraintes «dures», avec contrôle des zones d'influence. Cette dernière est abordée pour la résolution d'une contrainte de passage avant d'être déclinée, pour le cas de plusieurs contraintes, en deux méthodes itératives différentes. Nous montrons que ces méthodes peuvent se reformuler sous une forme matricielle et peuvent donc aussi se résoudre de façon algébrique. Nous introduisons ensuite une façon possible d'interpréter les contraintes dites «lâches» afin qu'elles soient exploitables dans notre moteur de résolution de contraintes dites «dures». Nous présentons, enfin, avant de conclure, notre moteur de résolution/construction qui exploite les outils précédemment cités.

4.2 Méthodes de déformation existantes

Il existe plusieurs méthodes de déformation permettant de modéliser des objets en trois dimensions. Dans cette étude, les techniques de déformation de modèles implicites et de modèles polyédriques ne sont pas traitées. Le lecteur pourra cependant se reporter à l'ouvrage de [Bloomenthal and al., 1997] ainsi qu'à la thèse de [Gesquiere, 2000] pour avoir plus d'informations concernant la manipulation des surfaces implicites. Il pourra aussi consulter des travaux portant sur la déformation de modèles polyédriques en se reportant aux travaux suivants [Parent, 1977, Borrel and Rappoport, 1994, Aubert and Bechmann, 1997,

Raffin, 2000], ou bien [Yoshizawa and al., 2002, Malik et al., 2002, Yoshizawa and al., 2003, Angelidis and al., 2004, Botsch and Kobbelt, 2004]. Nous ne nous intéressons dans cette partie qu'aux méthodes de déformation destinées à la manipulation des courbes et surfaces paramétriques. Comme nous l'avons vu précédemment, la forme des modèles paramétriques (Cf. 1.3.4) est contrôlée par un ensemble de pôles constituant le polyèdre (respectivement polygone) d'une surface (respectivement courbe). La déformation d'un tel modèle passe donc par la manipulation et le positionnement de ces points de contrôle dans l'espace. Pour cela, les techniques utilisées peuvent se répartir en trois groupes de déformations :

- les approches *géométriques* : les pôles sont déplacés de manière explicite en fonction de la déformation à appliquer ;
- les approches *physiques* et *mécaniques* : les déplacements appliqués aux pôles sont calculés de manière à ce que le comportement de la déformation semble le plus naturel possible ;
- les approches *multirésolutions* : le polyèdre de contrôle est décomposé en plusieurs niveaux de détail, avant de lui appliquer la déformation au niveau désiré. Le polyèdre est ensuite recalculé à son niveau de détail d'origine pour obtenir le modèle déformé.

Ces trois approches sont abordées dans les paragraphes suivants, mais il est possible de consulter les documents [Hahmann and Elber, 2004, Pernot, 2004] pour une étude plus approfondie de ces méthodes et pour leur comparaison.

4.2.1 Approches géométriques

Par ces approches, les pôles du polyèdre de contrôle sont déplacés de façon explicite en fonction de la déformation à appliquer. La déformation peut être réalisée en manipulant directement chacun des pôles, c'est ce que nous traitons dans la partie *manipulation directe*, ou bien de façon indirecte par l'intermédiaire d'un *volume de déformation*.

4.2.1.1 Manipulation directe

Bien que la manipulation des courbes et surfaces à pôles ne soit par réellement *directe* puisque leur manipulation passe par le déplacement de leurs points de contrôle, on qualifie ici de directe toute déformation agissant directement sur les pôles du modèle. Ainsi, le déplacement explicite des points de contrôle d'une courbe ou d'une surface est la méthode de déformation la plus directe des modèles paramétriques. Cependant pour réaliser des déformations complexes ou pour manipuler des surfaces comprenant de nombreux pôles, l'opération peut être longue et très fastidieuse. C'est pour cela que plusieurs travaux ont été entrepris dans le but de faciliter et d'assister la démarche de déformation. [Bartels and Beatty, 1989], par exemple, développe une technique permettant de déplacer de façon interactive un point d'une courbe comme si l'utilisateur agissait directement dessus. Ainsi cette méthode cache à l'utilisateur la manipulation des pôles bien que ce soient

eux qui soient directement déplacés. À travers cette technique transparait aussi la notion de contrainte, puisque le point tiré de la courbe n'est autre qu'un point de passage forcé. Cette notion est ensuite développée dans [Fowler and Bartels, 1993] qui propose une méthode permettant de résoudre simultanément plusieurs points de passage associés à des contraintes de tangence (Fig. 4.1). Ainsi l'utilisateur a la possibilité de donner des points dans l'espace et d'obtenir la courbe qui interpole l'ensemble de ces points tout en respectant un ensemble de tangences. Une extension de cette méthode sur les surfaces est proposée dans [Fowler, 1992]. En adaptant, entre autre, le système de résolution de [Fowler and Bartels, 1993], [Schichtel, 1993a] propose de déformer les courbes en respectant une contrainte ponctuelle tout en gardant une «allure» globale la plus proche de l'originale. Une extension aux surfaces est ensuite proposée dans [Schichtel, 1993b].

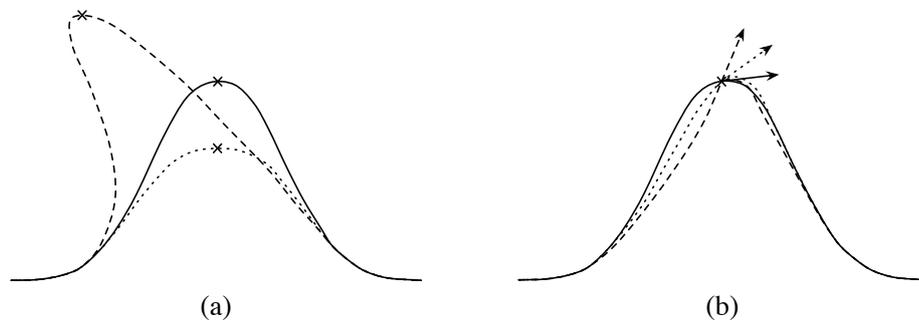


Figure 4.1. Interpolation : satisfaction d'une contrainte ponctuelle (a) et d'une contrainte tangentielle (b) [Fowler and Bartels, 1993]

Dans le cas d'un grand nombre de contraintes où il n'existe pas de solution interpolant tous les points de passage, il faut alors passer à des méthodes d'approximation [Hormann, 2001, Cohen and Merrien, 2003, Yang and al., 2004, Goldenthal and Bercovier, 2004]. Ainsi, le modèle considéré approche au mieux l'ensemble des points contraints. Pour résoudre ce problème, des méthodes de résolution aux moindres carrés [Hanson and Lawson, 1969, Borges and Pastva, 2002, Wang and al., 2005] sont utilisées (Fig.4.2).

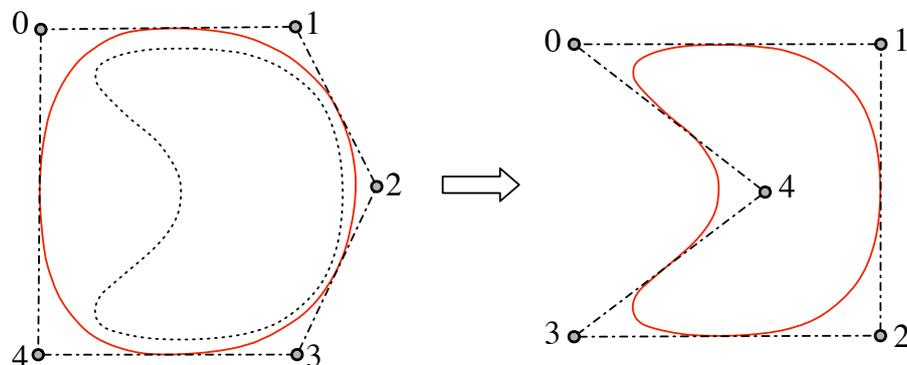


Figure 4.2. Approximation d'un ensemble de points par la méthode SDM (Squared Distance Minimization) [Yang and al., 2004]

Enfin, outre ces méthodes assistées permettant de manipuler la surface en appliquant des contraintes de passage, il existe des méthodes agissant directement sur le polyèdre de contrôle. Pour cela, il faut considérer ce polyèdre de contrôle comme un modèle polyédrique quadrangulaire, et ensuite le manipuler en lui appliquant des déformations spécifiques aux modèles polyédriques dont on a rapidement parlé en introduction du paragraphe 4.2. Ce type de déformation ne permet pas de satisfaire des contraintes précises, mais il donne à l'utilisateur des moyens efficaces de manipulation des pôles de la surface à sculpter. Plusieurs de ces déformations comme «*aplatir*» (Fig. 4.3), «*plier*» ou le *gauchissement*¹ surfacique [Blanc, 1994] sont présentées dans [Cobb, 1984, Piegl and Tiller, 1997].

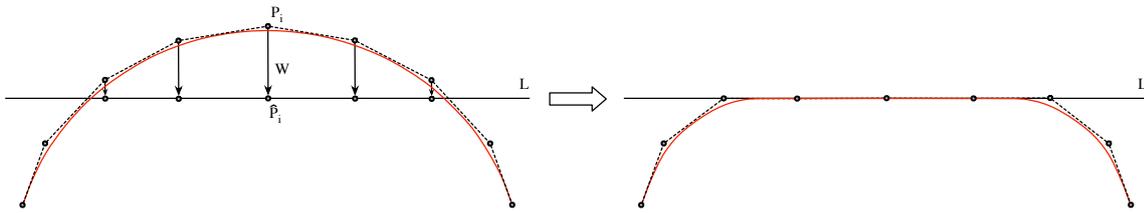


Figure 4.3. Aplatissement d'une courbe [Piegl and Tiller, 1997]

Une autre méthode de manipulation géométrique de la forme d'une surface (ou d'une courbe) est de plonger son polyèdre (ou polygone) de contrôle dans un volume de déformation. Cette technique et ses variantes sont rapidement expliquées dans la partie suivante.

4.2.1.2 Manipulation indirecte par volume de déformation

Les volumes de déformation sont habituellement connus sous la dénomination de FFD pour **Free-Form Deformation** ou déformation de forme libre. Cette technique permet de déformer globalement un objet en le plongeant dans un volume paramétrique à trois dimensions. Ce volume est la plupart du temps défini par un treillis associé à une base de Bézier ou B-Spline. Il est souvent fait l'analogie du volume paramétrique avec un bloc de plastique souple dans lequel l'objet à déformer serait plongé. Ainsi, en manipulant les pôles de ce treillis, c'est l'espace qu'il englobe qui est modifié et l'objet contenu dans cet espace suit ses déformations (Fig. 4.4).

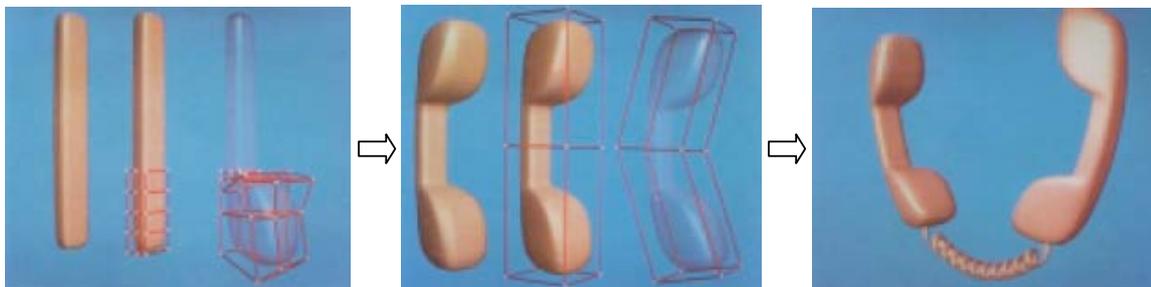


Figure 4.4. Modélisation par FFD [Sederberg and Parry, 1986]

¹de l'anglais «warping»

Les FFD sont introduites par [Sederberg and Parry, 1986], elles seront ensuite étendues en EFFD (pour **E**xtended FFD) par [Coquillart, 1990] où les treillis utilisés, jusque-là parallélépipédiques, peuvent être de forme quelconque (Fig. 4.5).

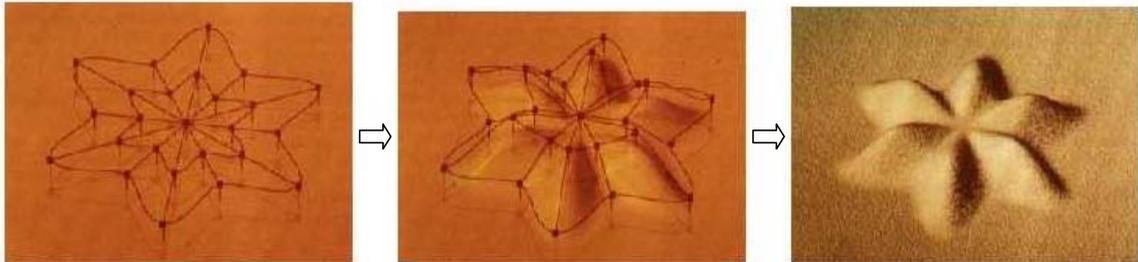


Figure 4.5. Exemple de déformation par EFFD [Coquillart, 1990]

Dans [Hsu and al., 1992], c'est une méthode permettant de manipuler directement l'objet plongé dans le volume paramétrique qui est présentée, ce qui permet à l'utilisateur de faire abstraction des pôles du treillis.

Les déformations axiales sont traitées dans [Lazarus and al., 1994] où l'utilisateur définit un *chemin* par une courbe B-Spline qu'il positionne sur l'objet à déformer. La manipulation de cette courbe déforme l'espace cylindrique qui l'entoure et la partie de l'objet incluse dans ce volume est déformée.

Sur une idée qui s'en approche mais en utilisant cette fois-ci des surfaces, [Feng and al., 1996] développe le concept de *surfaces d'influence*.

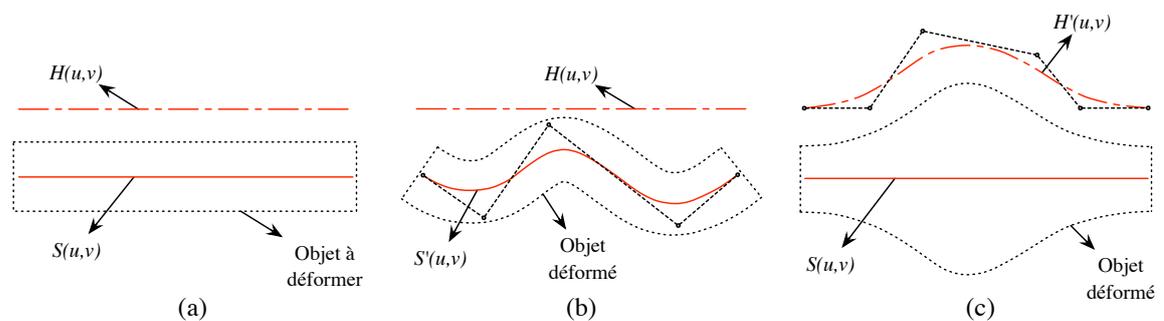


Figure 4.6. Déformation par surfaces d'influence [Feng and al., 1996]

La figure 4.6.a présente, en coupe, un exemple d'objet à déformer associé à sa *surface de forme* $S(u, v)$ et à sa *surface de hauteur* $H(u, v)$. Les figures 4.6.b et 4.6.c montrent respectivement une modification de la surface de forme qui agit sur morphologie de l'objet et une modification de la surface de hauteur qui agit, cette fois-ci, sur la surface de l'objet. Un peu plus tard, [Crespin, 1998] fait intervenir dans les FFD la notion de pondération par fonction d'influence sur la déformation. Il adapte à sa méthode le concept de déformation par axe courbe et par surface. La figure 4.7 illustre ces deux adaptations par un exemple.

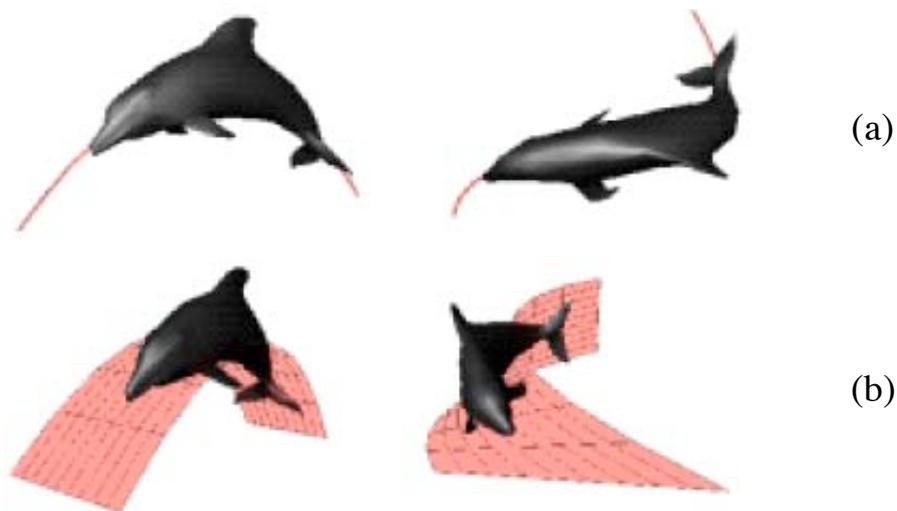


Figure 4.7. Déformation par axe courbe (a) et par surface quadratique (b) [Crespin, 1998]

La manipulation du treillis de la FFD implique dans la grande majorité des cas un changement de son volume, ce qui entraîne aussi des modifications du volume de l'objet à déformer. [Aubert and Bechmann, 1997] se penche sur ce problème et propose une méthode itérative basée sur DOGME² [Borrel and Bechmann, 1991] pour garder un treillis à volume constant lors de la déformation.

Enfin, [Debunne, 2000] introduit une version multirésolution des FFD. Cette approche permet de raffiner le treillis aux endroits désirés autorisant ainsi des déformations à la fois locales et globales de l'objet immergé dans le volume paramétrique. De nombreux autres travaux se rapportant aux FFD ont encore été réalisés, certains sont disponibles dans [Bechmann and al., 1997, Bechmann and Gerber, 2003] et [Raffin, 2000].

Bien que les déformations de forme libre soient des outils très puissants pour la déformation des modèles géométriques et pour leurs animations, ils ne sont pas très adaptés à la manipulation des courbes et surfaces à pôles. En effet, dans ce cas, c'est le polyèdre (ou polygone) de contrôle qui subit les déformations du volume paramétrique et qui les répercute ensuite sur la surface (ou la courbe) à pôles. Bien que cette technique reste tout à fait envisageable, cette double indirection est parfois difficile à appréhender pour l'utilisateur. Les FFD sont donc, pour cette raison, beaucoup plus utilisées dans la manipulation des modèles polyédriques. Dans le souci d'obtenir des déformations réalistes et plus naturelles, la manipulation des courbes et surfaces à pôles fait l'objet de travaux s'orientant vers les phénomènes physiques et des comportements répondant aux règles de la mécanique. Ces approches sont présentées dans le paragraphe suivant.

²Deformation Of Geometric Model Editor

4.2.2 Approches physiques et mécaniques

Les approches physiques et mécaniques ont été introduites en modélisation géométrique par [Terzopoulos and al., 1987]. L'idée proposée dans cet article est d'utiliser la théorie de l'élasticité pour modéliser le comportement des surfaces et objets non solides comme les textiles par exemple. D'un point de vue plus général, les approches physiques et mécaniques reposent sur l'idée d'une modélisation des déformations qui paraissent plus «naturelles» et plus proches du réel.

[Bloor and Wilson, 1990] introduit pour cela les B-Splines déformables qui se généralisent dans [Terzopoulos and Qin, 1994, Qin and Terzopoulos, 1996] en D-Nurbs pour *Dynamic NURBS*. Ce modèle est une généralisation du modèle géométrique NURBS au domaine de la physique. Il intègre, en effet, la notion de temps, de masse et d'énergie de déformation dans la formulation standard des NURBS. Le comportement dynamique de ce modèle résulte de l'utilisation de la dynamique Lagrangienne. La résolution de contraintes de passage a aussi été traitée pour ce modèle comme cela est illustré sur la figure 4.8.

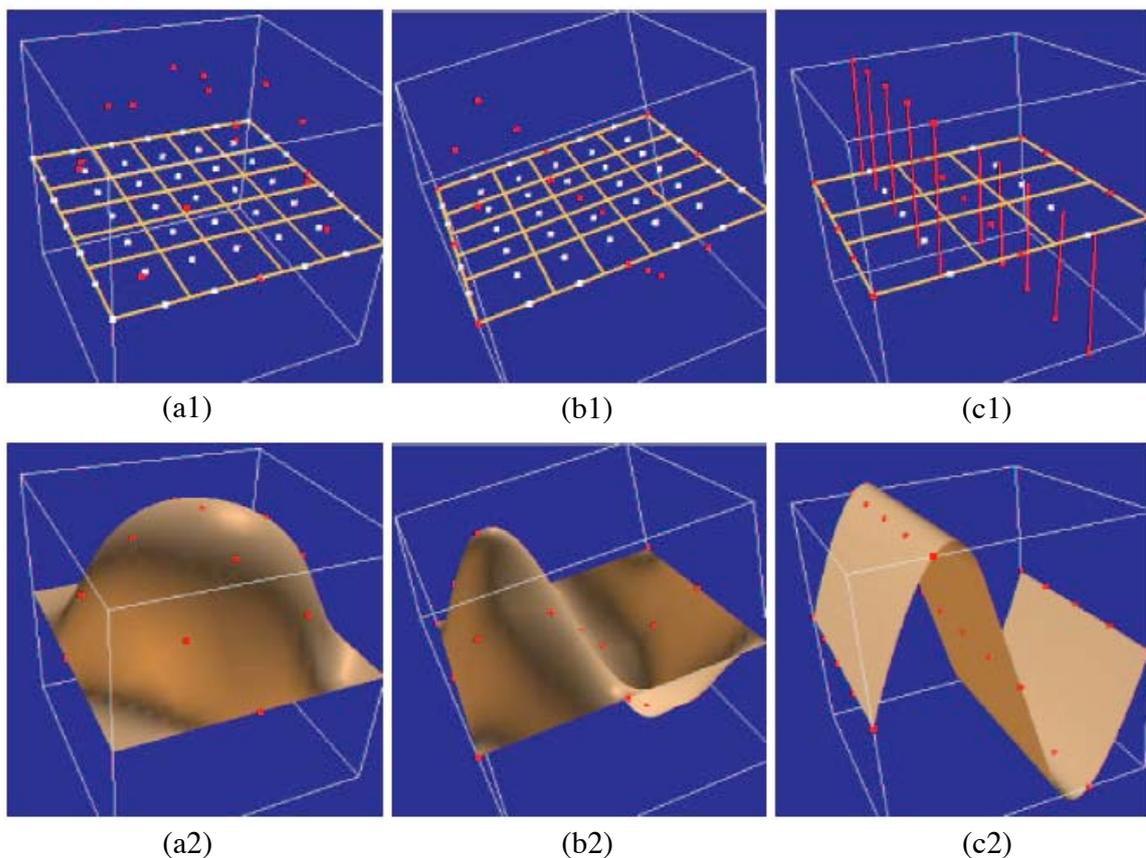


Figure 4.8. Ajustement optimal d'une Surface D-NURBS à un nuage de points issu d'un hémisphère (a), d'une forme convexe/concave (b) et d'une surface sinusoidale (c) [Qin and Terzopoulos, 1996]

Toujours dans le soucis d'être plus proche d'un comportement «naturel», une autre méthode consiste à restreindre les mouvements des pôles de la surface afin qu'ils suivent un ensemble de lois physiques. Dans [Thingvold and Cohen, 1990], les auteurs confondent le polygone de contrôle d'une surface B-Spline avec un modèle mécanique constitué de masses, ressorts et charnières (Fig. 4.9).

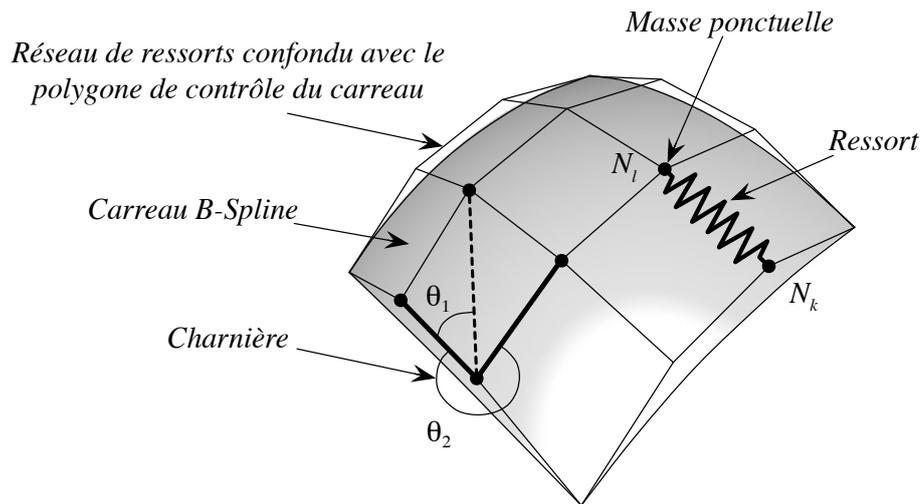


Figure 4.9. Modèle mécanique constitué de masses, ressorts et charnières [Guillet, 1999]

Chaque pôle de la surface coïncide avec un nœud N_k du réseau de ressorts et est associé à une masse. Les charnières permettent de relier quatre nœuds du réseau avec des degrés de liberté de 45° pour θ_1 et de 315° pour θ_2 . En contrôlant l'ensemble des paramètres physiques des composants du réseau, l'utilisateur peut indirectement contrôler la raideur de la surface B-Spline.

Enfin, toujours en assimilant le polyèdre de contrôle à une structure mécanique, l'équipe du professeur Jean-Claude Léon, du laboratoire 3S³ de Grenoble est partie des travaux de Schek [Schek, 1974] et les a adaptés aux modèles paramétriques de surfaces gauches [Léon and Trompette, 1995, Guillet, 1999, Lesage, 2002, Pernot, 2004]. L'idée consiste à confondre le polyèdre de contrôle de la surface avec un réseau de barres et d'en calculer la position d'équilibre statique. Les barres du réseau sont reliées entre elles en chaque nœud du polyèdre de contrôle par des liaisons rotules supposées parfaites. Les nœuds du système peuvent être fixes ou libres, ceux qui sont libres sont maintenus en équilibre par les efforts extérieurs qui leur sont appliqués. Chaque barre est associée à un paramètre mécanique q_i , strictement positif, appelé *densité de force*.

Pour déformer le carreau, il suffit alors de modifier soit sur la densité de force des barres, soit les forces extérieures agissant sur les nœuds du réseau. La position d'équilibre des nœuds

³Sols, Solides, Structures

est recalculée et le carreau déformé est obtenu. À travers ces recherches, les équipes des laboratoires 3S et I.M.A.T.I.⁴ de Gènes, développent un logiciel nommé DEFSURF dont la vocation est d'offrir aux utilisateurs des outils de déformation de surfaces gauches de haut niveau.

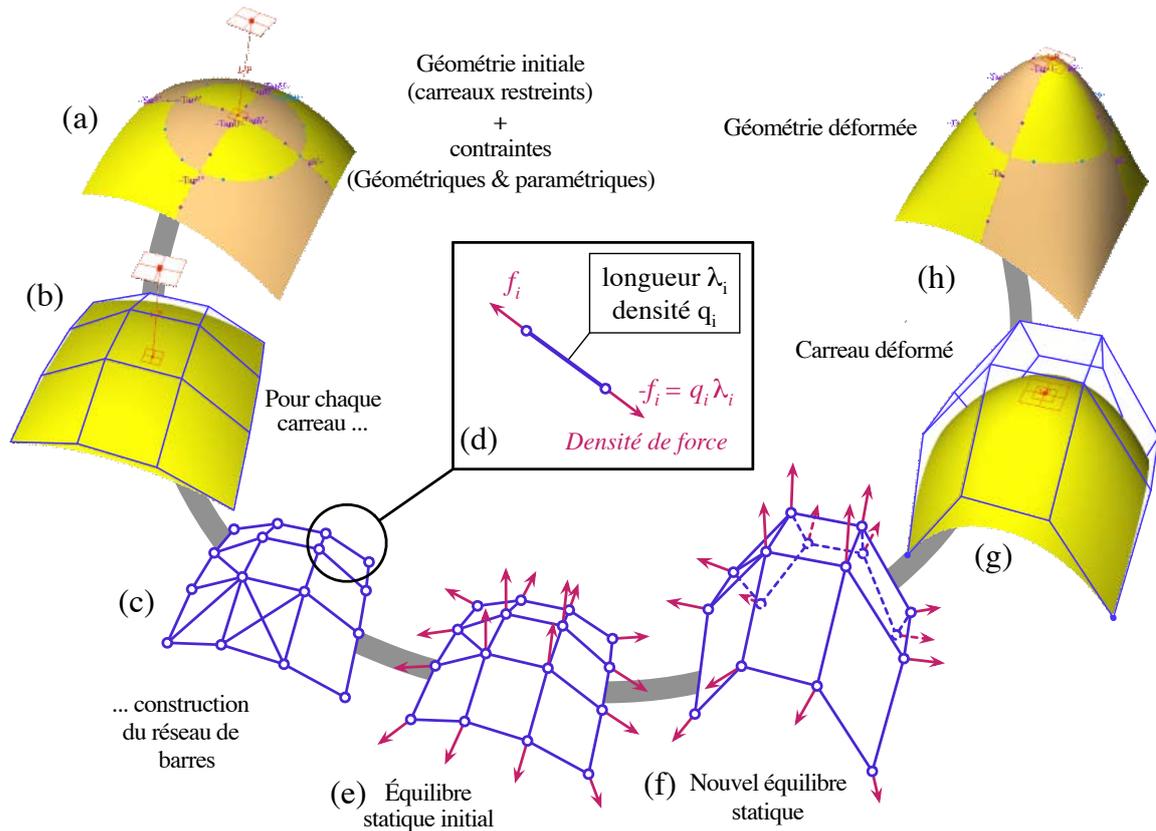


Figure 4.10. Méthode de déformation de surfaces gauches [Pernot and al., 2003]

Ainsi avec DEFSURF, il est possible de gérer des déformations agissant sur plusieurs carreaux de surface et satisfaisant des contraintes de passage par des points et des lignes tout en respectant des exigences de tangence à des plans. La figure 4.10 schématise le principe général de déformation utilisé à travers une contrainte de passage et une exigence de tangence à un plan.

Afin de ne demander à l'utilisateur qu'un nombre minimum de paramètres d'entrée sans toutefois le limiter dans sa démarche créative, DEFSURF offre au concepteur la possibilité de manipuler deux types de lignes de contraintes. Les premières sont appelées *lignes cibles*. Ce sont des courbes gauches par lesquelles la surface déformée devra passer. Elles permettent ainsi à l'utilisateur de donner l'allure qu'il souhaite localement obtenir sur la surface. Les secondes se nomment *lignes de limitation*, elles sont construites sur la surface afin de contrôler l'étendue de la déformation et conditionnent ainsi la forme de la surface

⁴Istituto di Matematica Applicata e Tecnologie Informatiche

déformée aux abords des lignes cibles. La figure 4.11 illustre le concept de ligne cible et de ligne de limitation à travers un exemple.



Figure 4.11. DEFSURF : Utilisation de lignes cibles et ligne de limitation [Pernot and al., 2003]

Ces lignes de contraintes sont discrétisées en plusieurs contraintes ponctuelles afin de recalculer la position d'équilibre du réseau de barres. Cependant, dans la majorité des cas, le système à résoudre est sous-contraint, il existe alors une infinité de solutions au problème. Pour n'obtenir qu'une solution, DEFSURF propose de résoudre le problème tout en minimisant une fonctionnelle. Chaque fonctionnelle proposée permet de mettre en avant une propriété particulière de la surface résultat comme par exemple la minimisation de son aire ou bien encore un changement de forme minimum. Ces fonctionnelles peuvent aussi être vues sous l'aspect de contraintes dites «lâches » que nous avons développé précédemment. La figure 4.12 présente quelques exemples de résultats de déformation sur un même objet avec les mêmes contraintes mais en utilisant différentes fonctionnelles.

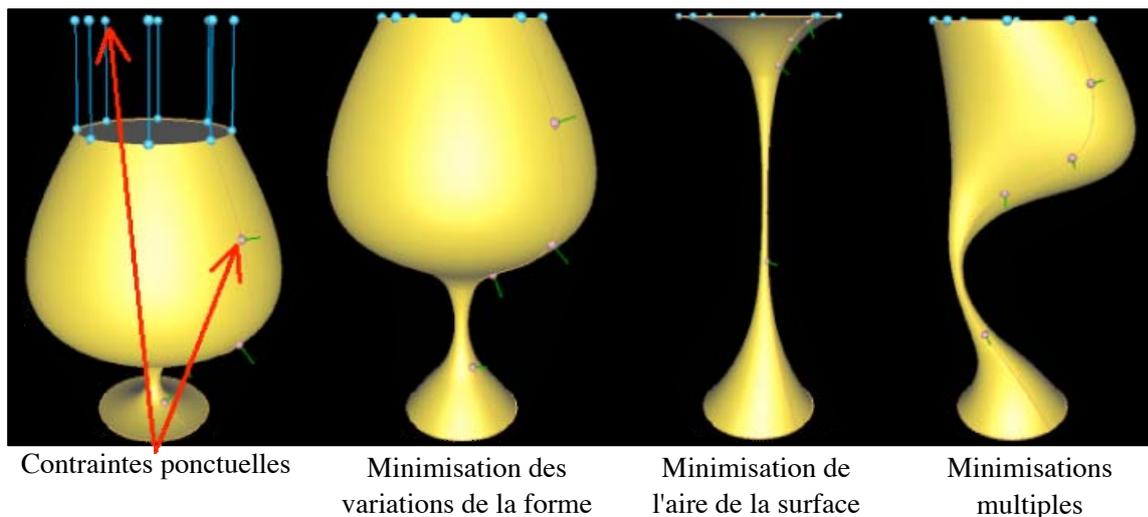


Figure 4.12. DEFSURF : Impact de la fonctionnelle sur la déformation [Pernot, 2004]

Outre les approches géométriques, physiques et mécaniques que l'on vient d'aborder, des travaux ont été menés dans le domaine de la multirésolution afin de décomposer une surface en plusieurs niveaux de détail. Ces représentations sont ensuite exploitées pour l'application de différentes déformations. Le paragraphe suivant présente ces différentes études.

4.2.3 Approches multirésolutions

Les approches multirésolutions font actuellement l'objet d'une attention toute particulière dans les domaines de l'informatique graphique tant à travers la modélisation géométrique que le traitement des images. La multirésolution a, en effet, beaucoup de points attractifs comme la visualisation et les traitements sous plusieurs niveaux de détail, ou bien encore la compression. Dans le cadre de la décomposition puis de la déformation de courbes et surfaces à pôles, de nombreux travaux se sont basés sur l'utilisation des ondelettes [Mallat, 1989, Stollnitz and al., 1996]. Par cette technique, le polyèdre (ou polygone) de contrôle d'une surface (ou courbe) à pôles peut se décomposer en plusieurs niveaux de détail [Kazinnik and Elber, 1997] en connaissant le moyen de passer d'un niveau à l'autre.

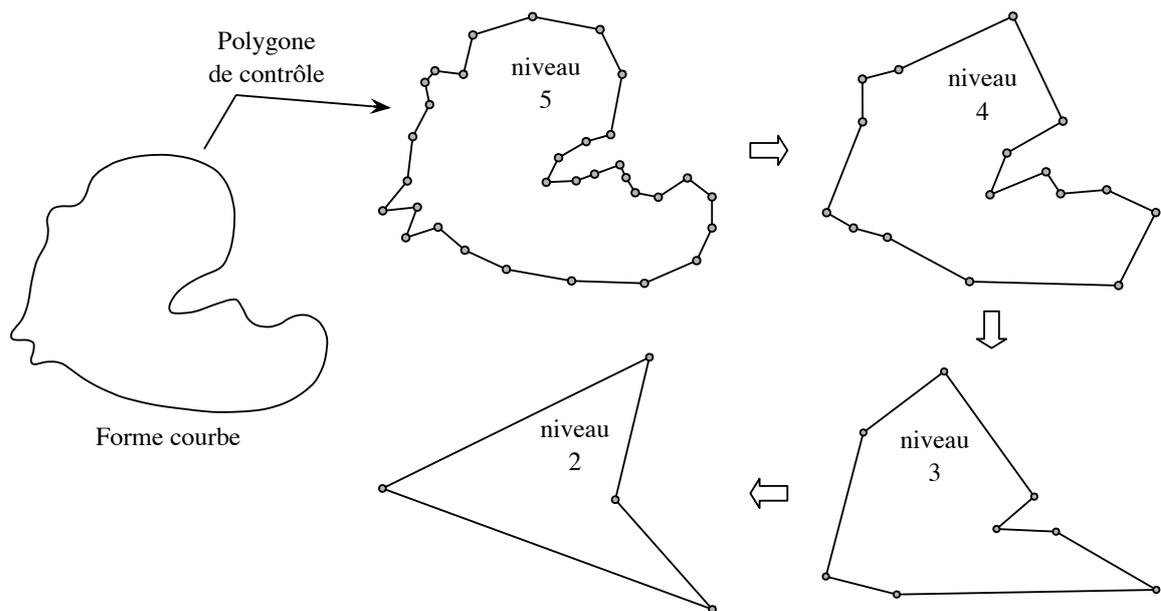


Figure 4.13. Courbe B-Spline définie par 32 pôles et les représentations de son polygone de contrôle en 5 niveaux de détails [Hahmann and al., 2005]

La figure 4.13 présente un exemple de décomposition d'un polygone de contrôle initialement constitué de 32 pôles puis représenté selon quatre autres niveaux de détail.

L'un des avantages de cette méthode est de pouvoir accéder à des polyèdres (ou polygones) de contrôle faits de peu de pôles, ce qui permet à l'utilisateur de faire des déformations plus ou moins globales en fonction du niveau de détail du polyèdre (ou polygone) manipulé. Une déformation en bas niveau de détail va, par exemple, consister à ne déplacer qu'un seul

pôle, ce qui est le cas sur la figure 4.14, mais cette déformation va être automatiquement répercutée sur tous les niveaux de détail supérieurs. Ainsi le déplacement d'un seul pôle est équivalent à la manipulation de plusieurs dizaines dans certains cas.

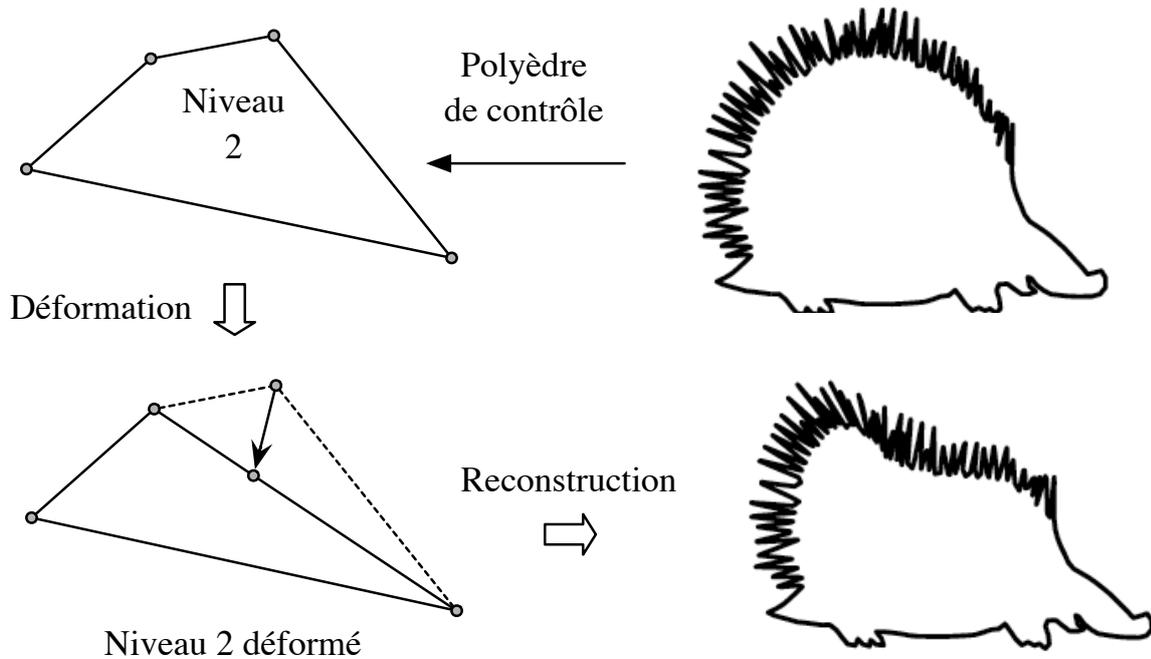


Figure 4.14. Déformation du polygone de contrôle de niveau détail faible et reconstruction de la courbe déformée en niveau de détail élevée

Enfin, une déformation dans un niveau de détail faible du polygone de contrôle ne va altérer, par cette technique, que la morphologie globale de l'objet et totalement préserver les détails de sa surface dont l'information est contenue dans les niveaux de détails supérieurs.

À cette technique, il est possible d'attacher d'autres fonctionnalités comme la résolution et le contrôle de contraintes par exemple. C'est ce qui est fait dans [Hahmann and Elber, 2004, Hahmann et al., 2005] où des contraintes de passage et des contraintes de tangence peuvent être satisfaites dans le cas d'une courbe à pôles. Le problème de la déformation de courbes planes fermées avec préservation de son aire a aussi été traité dans [Hahmann and al., 2005].

Après la présentation de toutes ces techniques de déformation existantes appliquées aux courbes et surfaces à pôles, nous détaillons, dans la partie suivante, les choix et les propositions de déformation que nous avons réalisés dans le but de construire les modèles géométriques associés à un modèle patron donné.

4.3 Méthodes de déformation proposées

À l'issue de l'étape de planification, le processus déclaratif que nous développons dans cette étude nous permet d'obtenir plusieurs modèles patrons issus d'un même modèle sémantique généralisé. Un modèle patron se différencie des autres par sa séquence de construction représentative d'une classe de solutions. Un modèle patron peut donc engendrer une infinité de modèles géométriques (Fig. 4.15), *instances* de la classe de solutions qu'il représente. L'objectif du moteur de résolution/construction est de construire un de ces modèles géométriques afin d'obtenir une solution géométrique qui puisse être présentée à l'utilisateur lors de la phase de prise de connaissance des solutions.

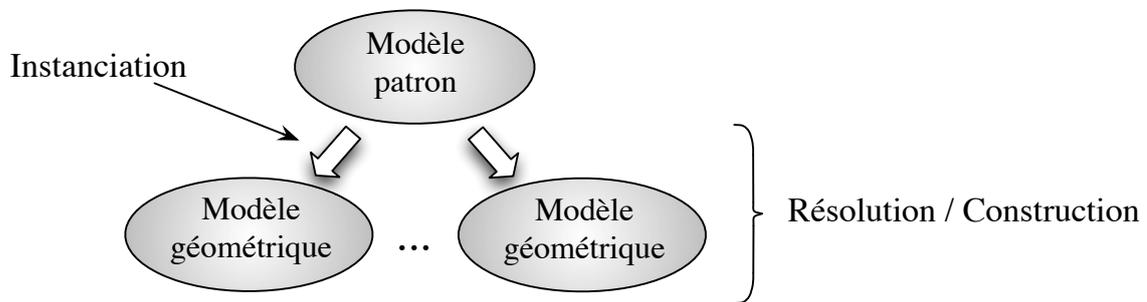


Figure 4.15. Arbre des modèles géométriques d'un modèle patron

Pour cela, nous proposons de partir d'une surface NURBS de degré 3×3 décrite par une grille de points de contrôle initialement positionnés dans le plan (O, \vec{x}, \vec{y}) de notre environnement. Cette grille est ensuite déformée selon la séquence de construction définie dans le modèle patron. Le moteur de résolution doit donc avant tout être un outil de déformation puissant de surfaces à pôles.

Nous nous sommes tout d'abord intéressés à l'utilisation du logiciel DEFSURF, présenté dans le paragraphe précédent. Comme nous l'expliquons dans la suite, cette piste a été abandonnée dans le cadre de cette thèse, mais reste une solution intéressante dans les perspectives de nos travaux. Nous nous penchons ensuite sur les techniques et méthodes de déformation que nous avons développées et qui reposent sur des approches géométriques.

4.3.1 Utilisation de DEFSURF

La première piste de travail étudiée pour obtenir des modèles géométriques à partir d'un modèle patron, s'est portée sur l'utilisation du logiciel DEFSURF. En effet, cet outil est tout spécialement développé pour déformer les courbes et les surfaces à pôles, il propose, de plus, de nombreuses possibilités de résolution de contraintes comme nous l'avons précisé dans la partie 4.2.2.

Dans ce logiciel, la déformation d'une surface nécessite la définition d'une ligne de limitation et d'une ou plusieurs lignes cibles. Ainsi il faut faire ressortir ces deux concepts de notre approche pour réaliser les déformations désirées. Les lignes de limitation se sont

tout naturellement dégagées puisque nous utilisons une notion très proche qui est celle de *zone*. Chaque zone est, dans notre approche, délimitée par une courbe NURBS dans le plan paramétrique du carreau de la surface traitée. La ligne de limitation d'une zone n'est donc que la représentation de cette courbe de l'espace paramétrique dans celui de l'espace tridimensionnel de la surface. La figure 4.16 illustre cette analogie.

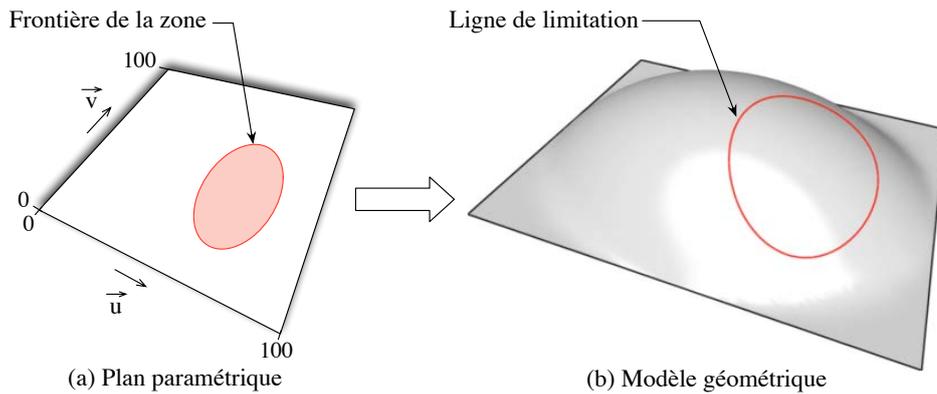


Figure 4.16. Analogie d'une ligne frontière de zone avec une ligne de limitation

La véritable difficulté réside dans la détermination de la ou des lignes cibles en fonction de la déformation désirée. Dans un cas très simple comme la déformation «Bombée» par exemple, il est possible de trouver des heuristiques plus ou moins complexes pour obtenir des résultats corrects. Ainsi, comme cela est illustré sur la figure 4.17, il est possible de voir la ou les lignes cibles comme une ou plusieurs lignes de niveau.

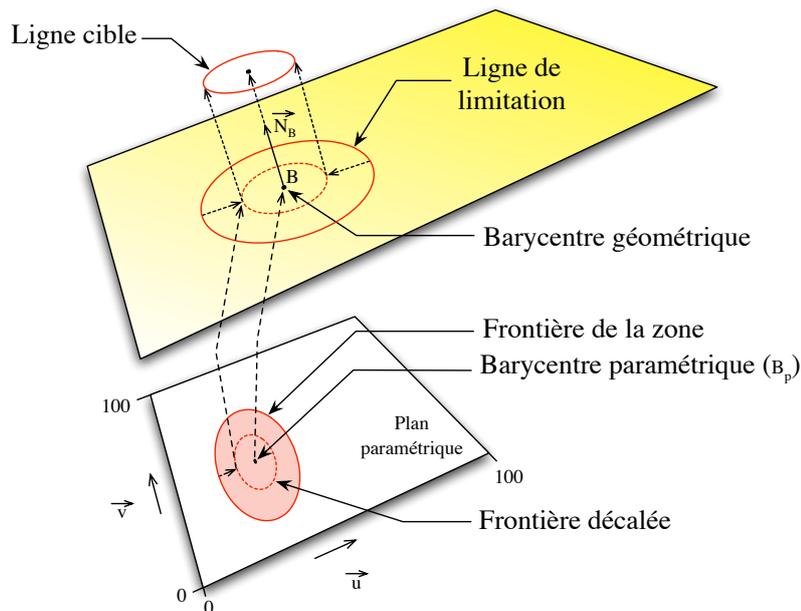


Figure 4.17. Construction possible d'une ligne cible pour la déformation «Bombée»

Pour cela, il faut déterminer, dans le plan paramétrique, une ou plusieurs frontières plus rétractées de la zone que l'on veut déformer. Une telle ligne, représentée en pointillé sur la figure 4.17, est décalée vers l'intérieur de la zone, on la nomme alors *frontière décalée*. Le barycentre paramétrique B_p de la zone, qui peut ne pas appartenir à cette zone si sa forme est concave, a son image sur la surface que l'on nomme barycentre géométrique B . L'image de la frontière décalée est, quant à elle, calculée sur la surface, puis translatée selon la direction de la normale en B . On obtient ainsi une ligne cible. Il est donc possible de contrôler deux paramètres pour définir une ligne cible de la déformation «Bombée» : le décalage entre la frontière de la zone et la frontière décalée, et la distance de translation selon la normale à B . La figure 4.18 montre deux résultats obtenus, d'abord sur une surface initialement plane en (a) puis sur une surface déjà bombée en (b).

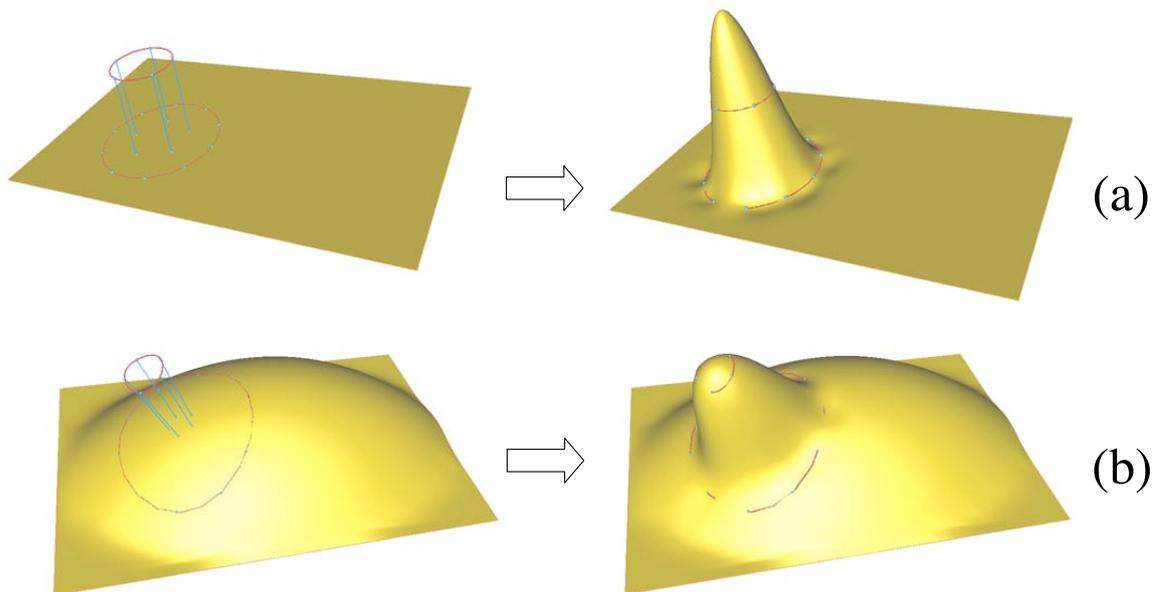


Figure 4.18. Déformation «Bombée» avec DEFSURF sur une surface plane (a) et sur une surface non plane (b)

Ces premiers résultats portant sur la déformation «Bombée» sont encourageants. Il existe, cependant, des cas délicats à manipuler pour toujours obtenir les effets escomptés. En effet, le concept de frontière décalée a des limites lorsque les frontières de la zone sont étroites. Il faudrait alors introduire la notion de *squelette* d'une zone comme étant la frontière décalée minimale pour traiter les cas limites.

Le second point délicat porte sur l'utilisation de la direction de la normale en B comme étant le support de translation de l'image, sur la surface, de la frontière décalée. Par cette méthode, c'est la forme locale au point B , qui peut ne pas appartenir à la forme à déformer, qui contrôle la position de la ligne cible. Or il serait plus cohérent de prendre en compte la forme globale de la zone pour obtenir des déformations plus intuitives. Ce problème,

assez classique, réapparaît dans la partie 4.3.4 où nous tentons de lui apporter une solution qui satisfasse à nos exigences à travers la notion de *vecteur normal moyen* traitée dans le paragraphe 4.3.4.2.

Bien qu'il soit possible de réaliser simplement des déformations comme «Bombée» ou «Creusée», il semble plus difficile de traiter des cas comme «Inclinée» ou «Courbée». D'un point de vue général, il est clair que l'utilisation d'un outil tel que DEFSURF est intéressant pour aboutir à des formes et à des modèles géométriques attendus par l'utilisateur. En outre, l'utilisation de fonctionnelles à minimiser permet de contrôler la résolution en respectant d'autres contraintes comme la minimisation d'aire qui sont essentielles dans des problèmes de CAO. Comme nous l'avons signalé dans le paragraphe 1.4.4 notre approche et celle proposée par les concepteurs de DEFSURF sont vouées à se rejoindre. Il reste cependant, de notre côté, un véritable travail de définition des déformations, en termes de contraintes et de fonctionnelles, à réaliser pour que ce lien soit efficace. Dans les délais restant pour la réalisation de la thèse, un tel travail n'est pas envisageable. C'est pourquoi cette étude est l'une des perspectives d'évolution prioritaire du processus déclaratif de surfaces. Néanmoins, pour pouvoir présenter des résultats et valider la démarche proposée en partant d'un modèle sémantique généralisé jusqu'à des modèles géométriques, nous avons entrepris de mettre au point des méthodes de déformation basées sur des approches géométriques. Nous commençons leurs descriptions dès le paragraphe suivant en présentant notre façon d'évaluer les quantificateurs qui seront utilisés durant toutes les étapes de déformation.

4.3.2 Évaluation des quantificateurs

Dans le modèle sémantique généralisé, un quantificateur (Cf. 3.2.3.4) se présente comme l'association de trois valeurs :

- *valeur moyenne* ($\mu \in [0, 100]$) ou espérance : valeur la plus adaptée à la description ;
- *écart type* ($\sigma \in \mathbb{R}^+$) : variation ($\pm 3\sigma$) possible de cette valeur ;
- *poids* ($w \in \mathbb{R}^+$) : accentuation ou, au contraire, adoucissement de la quantification faite par l'utilisateur. Ce poids évolue directement avec le nombre de «très» donné par l'utilisateur, nous choisissons de dire que $w = 0.5 \times NbTres$.

La fonction d'évaluation d'un quantificateur doit tout d'abord tirer aléatoirement une valeur Q du quantificateur comprise entre $[\mu - 3\sigma, \mu + 3\sigma] \in [0, 100]$ comme cela a été précisé dans la note 8 de la partie 3.2.3.4. Ce tirage doit respecter la densité de probabilité qui est rattachée au quantificateur (Fig. 3.12). Il suffit ensuite de prendre en compte l'influence du *poids* w . Cette influence doit être sans effet dans le cas d'un poids nul ou d'une valeur moyenne de $Q = 50\%$. En effet, la valeur 50% correspond à la sémantique *moyenne* qui est équivalente à *très moyenne*, *très très moyenne* et ainsi de suite. L'influence doit par contre avoir un effet de diminution dépendant de w si $Q \in [0, 50[$ et un effet d'accroissement dépendant de

w si $Q \in]50, 100]$. Dans tous les cas, la quantification finale doit rester positive ou nulle. Il faut donc définir une fonction $\mathcal{F}_w(Q)$ telle que :

$$\mathcal{F}_w : [0, 100] \mapsto \mathbb{R}^+$$

La fonction \mathcal{F}_w que nous choisissons est la suivante :

$$\mathcal{F}_w(Q) = \frac{1}{100} \frac{Q^{w+1}}{50^w} \quad (4.1)$$

La fonction \mathcal{F}_w est divisée par cent pour être ramenée entre 0 et 1 dans le cas où l'utilisateur ne réalise pas d'insistance, et au delà s'il évoque un appui particulier. La figure 4.19 présente la variation de la fonction de quantification selon les différentes valeurs prises par w .

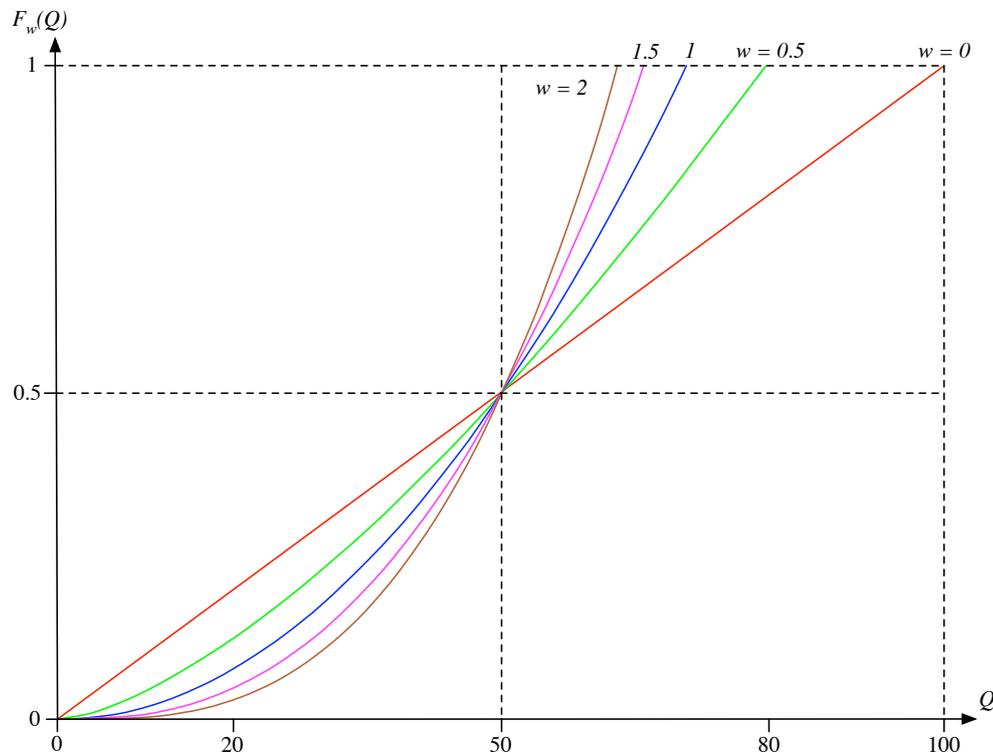


Figure 4.19. Variation de la fonction de quantification selon la valeur du poids

Dans toute la suite de ce document, lorsque nous parlons de la valeur du quantificateur, nous faisons allusion au résultat de cette fonction de quantification $\mathcal{F}_w(Q)$. La méthode de déformation commence alors par la préparation du polyèdre de contrôle dont nous proposons une technique de raffinement des zones à manipuler dans le paragraphe suivant.

4.3.3 Raffinement des régions à déformer

Travaillant sur des surfaces à pôles, la déformation d'une zone, ou région, de cette surface passe par le déplacement des sommets de son polyèdre de contrôle. C'est pourquoi il faut que celui-ci soit assez finement découpé pour pouvoir être suffisamment manipulé.

Nous choisissons de partir d'une surface initialement plane définie par une grille de contrôle plane. [Chambelland, 2001] propose quelques pistes de travail que nous reformulons :

- *l'approche totale* : le polyèdre de contrôle de la surface est très dense. Il contient suffisamment de pôles pour que toutes les contraintes et les déformations soient réalisées ;
- *l'approche constructive* : à l'étape initiale de la construction, nous considérons un polyèdre de contrôle constitué de $(p + 1) \times (q + 1)$ sommets coplanaires, où p est le degré de la surface selon la direction u et q le degré de la surface selon le degré v . Nous considérons par défaut que $p = q = 3$. Les étapes suivantes consistent à ajouter localement et régulièrement des lignes et des colonnes de pôles aux polyèdres afin de satisfaire les contraintes de déformation désirées. Cette approche a l'avantage de pouvoir maîtriser le nombre de pôles de la surface ;
- *l'approche par élimination de points* : cette méthode est la duale de la précédente, on considère un polyèdre très dense. Les étapes suivantes consistent à éliminer les points de contrôle qui ne sont pas nécessaires aux déformations.

Nous choisissons pour notre méthode une approche constructive, qui semble être la plus adaptée à notre problème tout en ayant un certain contrôle du nombre de pôles. Pour cela, les étapes suivantes sont réalisées :

1. Génération d'une surface plane contrôlée par $(p + 1) \times (q + 1)$ pôles, soit 4×4 points de contrôle par défaut, placés dans un plan. Pour fixer un référentiel, la position de ces pôles est initialement définie de façon régulière dans le plan (\vec{x}, \vec{y}) entre le point $(0,0,0)$ pour le pôle bas-gauche et le point $(3,3,0)$ de l'espace de travail pour le pôle haut-droit de la grille de contrôle. L'utilisateur est, quant à lui, considéré comme étant placé sur l'axe \vec{z} , dans ces valeurs positives, et regardant vers l'origine du repère de l'espace de travail. Le choix de ces positions donne une taille initiale à la surface. Les coefficients donnés pour les déformations qui sont traitées dans ce chapitre, sont définis pour donner de bons résultats à cette échelle. Le vecteur de nœuds associé à chaque direction de cette surface est défini selon des intervalles réguliers entre les nœuds et avec une multiplicité aux bords égale à $p + 1$ pour la direction u et $q + 1$ pour la direction v ;

Remarque 4.3.1 – *Il est important de noter que nous ne gérons pas les contraintes de dimensions. Une telle contrainte permettrait, par exemple, de donner à l'utilisateur la possibilité de spécifier la longueur exacte de la surface qu'il souhaite obtenir. Cette fonctionnalité est une perspective de travail pour l'évolution de notre étude qui n'a pas encore été abordée.*

2. Prise en compte de la taille et de la forme de la surface en fonction des données *forme* issues du modèle sémantique de la zone racine. Ainsi, si la surface est décrite comme *longue*, les pôles de la surface sont déplacés dans son plan (Fig. 4.20) en respectant les caractéristiques de la propriété «Longue» contenues dans la base de connaissances.

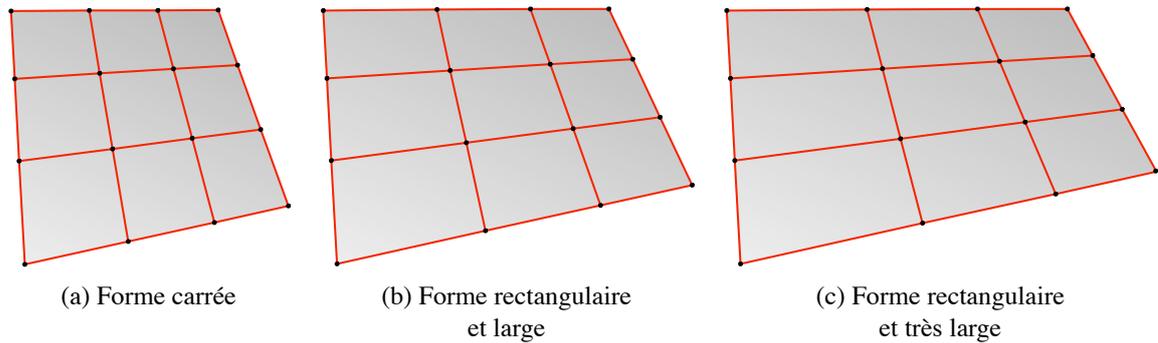


Figure 4.20. Surface plane avec prise en compte des propriétés de la forme

Ces déplacements sont ajustés avec la fonction de quantification vue à la partie 4.3.2. Nous gérons toutes les formes de surfaces dont les frontières ne sont pas complètement confondues avec les bords de l'espace comme des restrictions. Ainsi, dans le cas d'une surface *ronde*, *ovale* ou d'une forme quelconque, la frontière de la forme devient une courbe de restriction appliquée au plan paramétrique. La surface voit alors sa forme modifiée comme cela est illustré sur la figure 4.21 ;

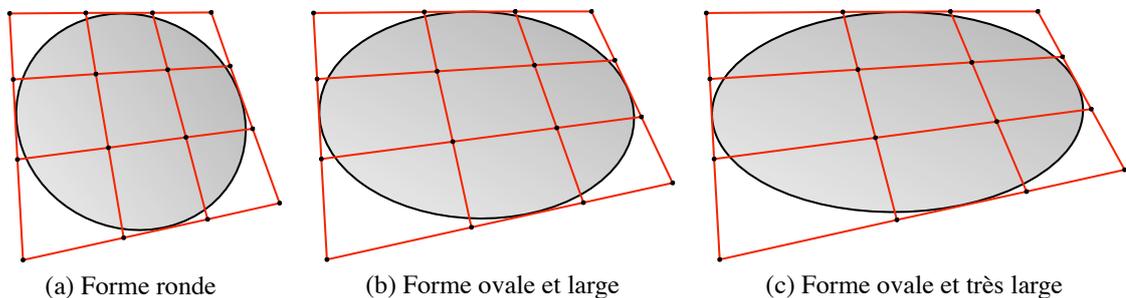


Figure 4.21. Surface plane avec prise en compte des restrictions de forme

3. Cette étape consiste à ajouter des lignes et des colonnes de pôles sur la grille de contrôle dans les régions correspondant aux zones à déformer. Pour y parvenir, nous utilisons la méthode de raffinement proposée dans [Boehm and Prautzsch, 1985] qui consiste à ajouter des nœuds dans les vecteurs de nœuds des directions u et v , faisant ainsi apparaître implicitement des lignes et des colonnes de pôles dans les régions correspondantes du polyèdre de contrôle. De plus, le raffinement des vecteurs de nœuds permet de mieux conserver la paramétrisation globale de la surface. Le choix de la position des nœuds à insérer dans les vecteurs de nœuds se fait à partir de la définition des zones dans le plan paramétrique. Afin d'être sûr de bien raffiner toute une zone, on raffine le lieu paramétrique défini par sa boîte englobante. Dans chacune des directions, les nœuds sont insérés par dichotomie jusqu'à ce qu'un seuil de raffinement soit atteint. Ce seuil, noté R est défini comme étant l'écart paramétrique

entre deux nœuds est choisi en fonction de la déformation à réaliser et en fonction de la taille de la zone sur laquelle on travaille. La base de connaissances contient en effet, pour chaque déformation, un raffinement minimum permettant d'obtenir des résultats visuellement corrects des déformations. Cette valeur comprise entre 0 et 100 a été déterminée pour une surface de degré 3×3 sur une moyenne de nombreuses réalisations d'une même déformation comme étant celle qui donne les *meilleurs*⁵ résultats visuels. Cette valeur de raffinement est ajustée à l'échelle de la zone Z_i à raffiner (R_i) afin d'être rapportée à l'échelle de la zone racine Z_0 (le plan paramétrique étant équivalent au repère global R_0). La valeur obtenue est considérée comme étant le seuil R de raffinement de la zone.

La figure 4.22 présente un exemple du raffinement de deux zones d'une même surface. Pour une meilleure visibilité, le polyèdre de contrôle de la surface n'est pas représenté sur cette figure et seuls les pôles sont visibles.

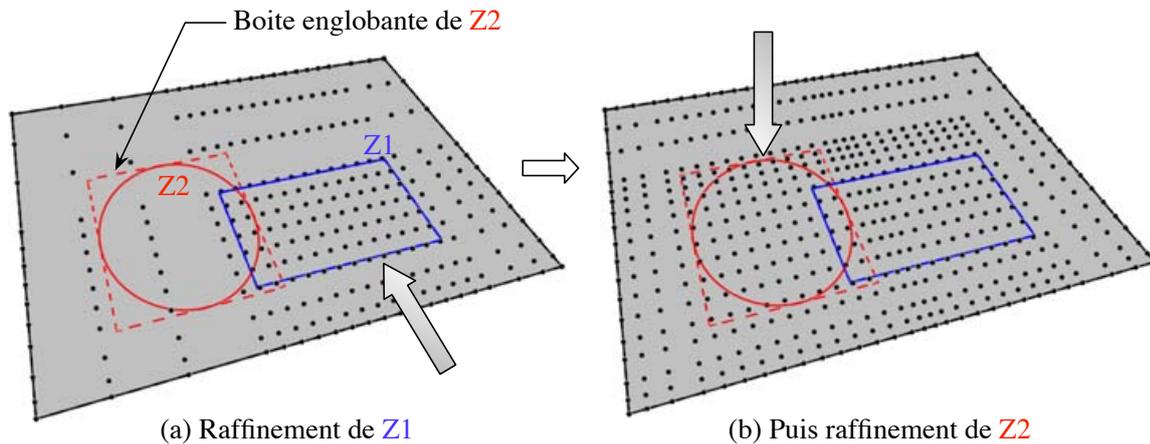


Figure 4.22. Raffinement de plusieurs zones d'une même surface

Pour appliquer cette technique, nous avons conçu l'algorithme 1 qui est consultable en annexe D.1 ;

4. Association des pôles aux zones à déformer. Le polyèdre de contrôle étant plan, rectangulaire et exclusivement constitué de carreaux eux aussi rectangulaires, il est possible de très rapidement connaître le couple de paramètre (u, v) pour lequel chaque pôle a le plus d'influence sur la surface. Pour cela nous utilisons la méthode dite de la «distance corde» donnée par [Piegl and Tiller, 1997] dans la partie traitant de l'interpolation globale. Nous nommons ce point l'*image paramétrique* d'un pôle. À partir de cette *représentation paramétrique* de la grille de contrôle, nous associons à chaque zone tous les pôles dont l'image paramétrique est incluse dans la courbe frontière de la zone considérée.

⁵ ceci n'est qu'une heuristique qui reste très dépendante de la subjectivité de l'utilisateur

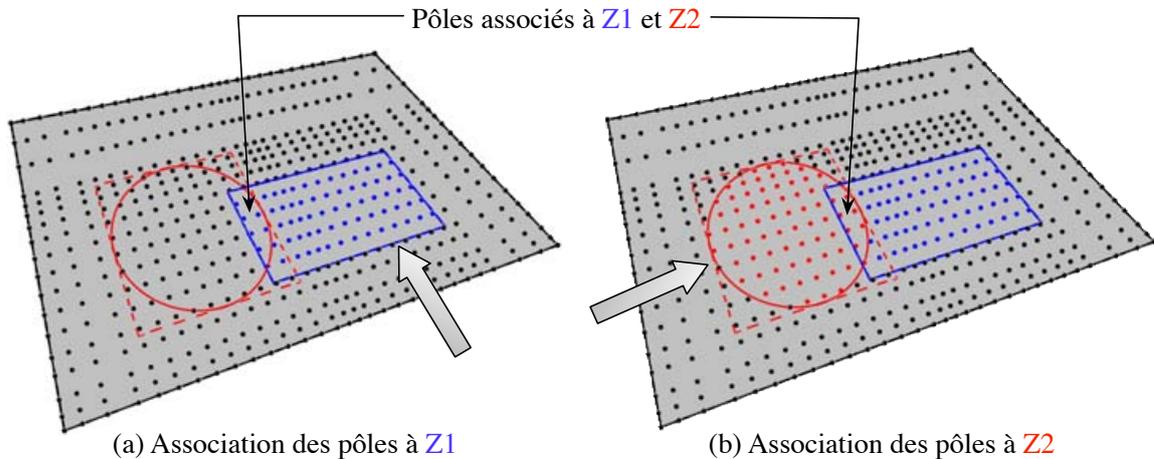


Figure 4.23. Association des pôles influents à chaque zone de la surface

Si plusieurs zones se chevauchent, il est possible que celles-ci partagent un certain nombre de points de contrôle.

Cette association ne permet pas de donner une information précise sur la localisation de l'image paramétrique du pôle par rapport à la zone considérée. Or nous aurons besoin de cette information pour déformer les zones. Ainsi, nous choisissons d'associer à chaque zone une grille, que l'on nomme *grille de distances paramétriques* ou *zone de contrôle*. Chaque case de cette grille représente l'image paramétrique d'un pôle ayant une influence sur la boîte englobante de la zone considérée. On associe alors à chacune de ces cases la distance paramétrique minimale de l'image paramétrique du pôle représentant la case, à la frontière de la zone traitée. Les images paramétriques des pôles qui ont une influence sur la boîte englobante mais qui sont en dehors des frontières de la zone sont considérées comme appartenant à la frontière elle-même (distance nulle à la frontière). Ainsi seuls les points de contrôles ayant une influence maximale dans la zone paramétrique sont pris en compte.

Pour cette opération, la distance que nous utilisons dans le plan paramétrique est la distance euclidienne. De plus, nous normalisons la grille afin de ne manipuler que des valeurs comprises entre 0 et 1. Pour illustrer la notion de zone de contrôle, la figure 4.24 présente les grilles de distances associées à deux zones d'une même surface. Les pôles noirs sont ceux qui sont indépendants de la zone considérée, les pôles verts sont ceux qui sont en dehors des frontières de la zone courante mais qui ont une influence sur sa boîte englobante (considérées comme appartenant à la frontière de la zone : distance nulle). Enfin le dégradé de la couleur associée aux pôles va du vert au rouge en passant par le jaune. Elle représente la distance euclidienne paramétrique de l'image paramétrique du pôle à la frontière intérieure de la zone. Le rouge correspond à la distance la plus grande (1) et le vert à la distance la plus faible (0).

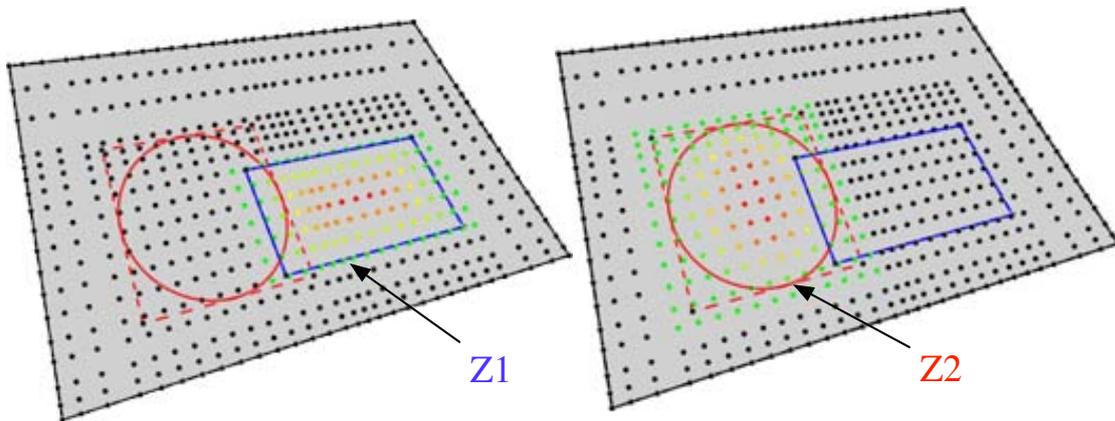


Figure 4.24. Les zones de contrôle d'une surface

Quelle que soit la position ultérieure des points de contrôle, ces zones de contrôle restent valides puisqu'elles sont définies à partir des données paramétriques de la surface. Au plus la distance est grande, c'est-à-dire au plus la couleur est rouge sur les figures, au plus on s'approche du *noyau* de la zone. Bien que cette information de distance ne tienne pas compte du poids attaché à chaque pôle, elle permet de positionner un point de contrôle à l'intérieur d'une zone. Dans le cas de zones dites trouées, les bords des trous sont considérés comme des frontières de la zone et le principe de distances est le même que précédemment. La figure 4.25 présente ce dernier cas.

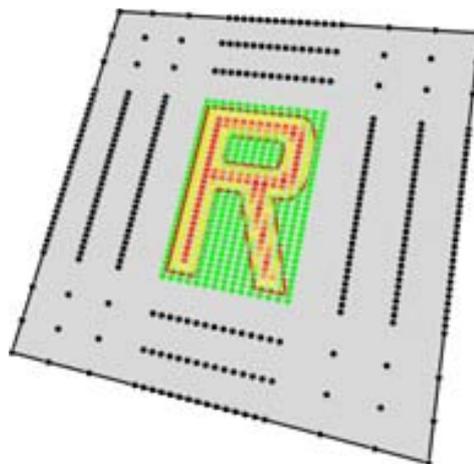


Figure 4.25. Cas d'une zone de contrôle trouée sur la surface

À l'issue de ces quatre étapes, le polyèdre de contrôle, encore plan, a la forme et la taille voulues et contient assez de points de contrôle pour pouvoir être déformé dans les zones désirées. Nous proposons donc dans le paragraphe suivant des déformations possibles.

4.3.4 Déformation par zone : contraintes «lâches»

Les contraintes dites «lâches» sont des allures que souhaitent voir apparaître l'utilisateur sur des zones de la surface résultat. Ainsi, la précision consistant à respecter exactement les frontières des zones ou les directions de déplacement ne sont pas exigées sauf dans le cas où le concepteur le spécifie. Dans ce dernier cas les contraintes sont dites «dures» et font l'objet de la partie suivante. Pour donner ces allures à la surface, c'est-à-dire appliquer les contraintes lâches, nous proposons et décrivons dans cette partie, un ensemble d'heuristiques. Ces heuristiques jouent alors le rôle d'opérateurs de déformation. Plusieurs d'entre elles utilisent les notions de *vecteur normal à un pôle*, de *vecteur normal moyen* ou de *plan moyen*. Nous décidons donc de les introduire avant de véritablement parler de déformations.

4.3.4.1 Vecteur normal à un point de contrôle

La notion de *vecteur normal à un point de contrôle* est un problème délicat qui se rapproche du calcul de la normale d'un sommet dans un modèle polyédrique. Le polyèdre de contrôle est un maillage quadrangulaire. Soit P_0 le pôle dont on veut calculer la normale, et P_1, P_2, P_3, P_4 ses quatre voisins comme cela est illustré sur la figure 4.26.

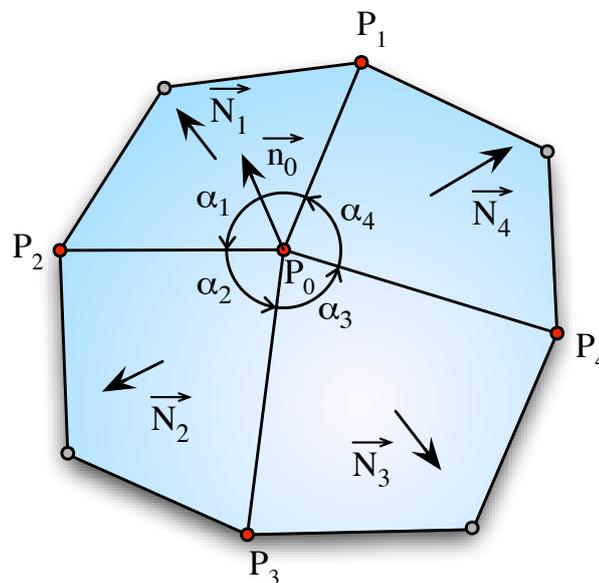


Figure 4.26. Normale à un pôle

Il est possible de calculer les normales $\vec{N}_1, \vec{N}_2, \vec{N}_3$ et \vec{N}_4 en considérant les quatre faces triangulaires voisines à P_0 comme suit (où \vec{N}_1 est la normale de la face triangulaire définie par (P_0, P_1, P_2) , \vec{N}_2 la normale de la face triangulaire définie par $(P_0, P_2, P_3) \dots$) :

$$\left\{ \begin{array}{l} \vec{N}_1 = \frac{\overrightarrow{P_0P_1} \wedge \overrightarrow{P_0P_2}}{\|\overrightarrow{P_0P_1} \wedge \overrightarrow{P_0P_2}\|_2} \\ \vec{N}_2 = \frac{\overrightarrow{P_0P_2} \wedge \overrightarrow{P_0P_3}}{\|\overrightarrow{P_0P_2} \wedge \overrightarrow{P_0P_3}\|_2} \end{array} \right. \quad \left\{ \begin{array}{l} \vec{N}_3 = \frac{\overrightarrow{P_0P_3} \wedge \overrightarrow{P_0P_4}}{\|\overrightarrow{P_0P_3} \wedge \overrightarrow{P_0P_4}\|_2} \\ \vec{N}_4 = \frac{\overrightarrow{P_0P_4} \wedge \overrightarrow{P_0P_1}}{\|\overrightarrow{P_0P_4} \wedge \overrightarrow{P_0P_1}\|_2} \end{array} \right.$$

Il n'existe pas de formulation fixée permettant de calculer la normale d'un sommet dans un maillage discret. Nous choisissons alors de déterminer \vec{n}_0 , normale à P_0 , comme étant la moyenne des quatre normales de ses faces voisines, pondérée par la valeur absolue de l'angle que partage chaque face avec P_0 [La Greca, 2001]. Le calcul est le suivant pour P_0 :

$$\vec{n}_0 = \frac{\sum_{i=1}^4 |\alpha_i| \vec{N}_i}{\sum_{i=1}^4 |\alpha_i|} \quad \text{puis} \quad \vec{n}_0 = \frac{\vec{n}_0}{\|\vec{n}_0\|_2} \quad \text{pour normaliser } \vec{n}_0 \quad (4.2)$$

Cette normale doit être mise à jour à chaque changement de position d'un des 4-voisins du pôle considéré.

4.3.4.2 Vecteur normal moyen à plusieurs points de contrôle

Le concept de *vecteur normal moyen* a été développé dans l'idée de tenir compte de la forme locale existante au niveau d'une zone avant de réaliser sa déformation. Ainsi, le vecteur normal moyen \vec{n} n'est autre qu'un vecteur normé indiquant la direction qui semble être la plus présente parmi celles indiquées par l'ensemble des normales des pôles attachés à la zone \mathcal{Z} qui va être déformée. Pour le calculer, nous définissons \bar{P} comme l'ensemble des \bar{N} pôles associés à \mathcal{Z} . w_i est le poids du pôle \bar{P}_i et \vec{n}_i est son vecteur normal. Nous calculons alors la normale moyenne de \mathcal{Z} comme la moyenne pondérée de toutes les normales des pôles de \bar{P} :

$$\vec{n} = \frac{\sum_{i=0}^{\bar{N}-1} w_i \vec{n}_i}{\sum_{i=0}^{\bar{N}-1} w_i} \quad \text{puis} \quad \vec{n} = \frac{\vec{n}}{\|\vec{n}\|_2} \quad \text{pour normaliser } \vec{n} \quad (4.3)$$

4.3.4.3 Plan moyen à plusieurs points de contrôle

Nous appelons *plan moyen* à un ensemble \bar{P} de pôles, le plan orthogonal au vecteur normal moyen \vec{n} des points de contrôle de \bar{P} passant en moyenne au plus près de chacun de ces points. En pratique, soit \mathcal{D} la droite de support \vec{n} . On projette orthogonalement les pôles de \bar{P} sur \mathcal{D} et on fait la moyenne pondérée (par le poids de chaque pôle) des positions des images projetés. Le plan moyen \mathcal{P} de \mathcal{Z} est alors le plan orthogonal à \vec{n} passant par le point résultat de cette moyenne. La figure 4.27 illustre ce traitement sur une courbe pour une meilleure vision du procédé.

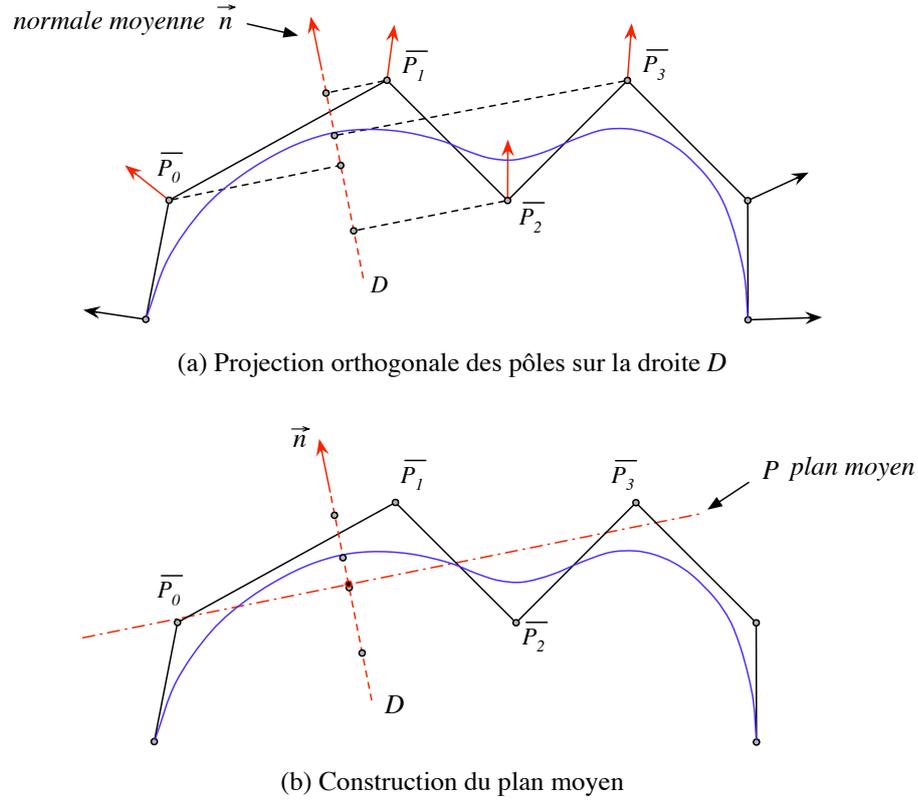


Figure 4.27. Construction du *plan moyen* à un ensemble de pôles

Ces trois outils étant introduits, nous pouvons présenter les déformations de zones que nous proposons en commençant par le traitement de *tension* et de *lissage*.

4.3.4.4 Aplanir, Tendre ou Lisser une zone

Nous considérons que l'action d'*aplanir*, de *tendre* ou de *lisser* une zone de la surface consiste à diminuer l'énergie de déformation attachée à cette zone. Pour cela nous réalisons un processus itératif qui consiste, à chaque itération, à déplacer les pôles attachés à une zone à la place des points sur la surface correspondant à leur image paramétrique. En d'autres termes, si $I\bar{P}_i$, associé à (u_i, v_i) dans le plan paramétrique, est l'image paramétrique du pôle \bar{P}_i , alors le déplacement appliqué à \bar{P}_i lors de chaque itération est : $\vec{d}_i = \overrightarrow{\bar{P}_i I\bar{P}_i}$. En répétant successivement cette opération, la méthode a pour effet de lisser la zone jusqu'à ce que les pôles soient confondus avec la surface. Nous considérons qu'au plus la somme SD des normes des déplacements apportés aux pôles durant une itération est faible, au plus nous nous rapprochons d'une surface tendue. Ainsi notre condition d'arrêt du processus est une heuristique consistant à comparer SD avec une fonction dépendant du quantificateur $\mathcal{F}_w(Q) \in \mathbb{R}^+$ donné dans le modèle sémantique généralisé. Nous choisissons cette fonction comme étant :

$$a(\mathcal{F}_w(Q)) = \begin{cases} \left(\frac{100-\mathcal{F}_w(Q)}{100}\right)^2 & \text{Si } \mathcal{F}_w(Q) \in [0, 100[\\ \varepsilon & \text{Sinon} \end{cases}$$

ε est ici considéré comme une valeur très petite et le processus itératif est réalisé tant que $SD > a(\mathcal{F}_w(Q))$. La figure 4.28 présente une surface initialement bombée en son centre sur laquelle on applique trois niveaux de lissage avec cette méthode.

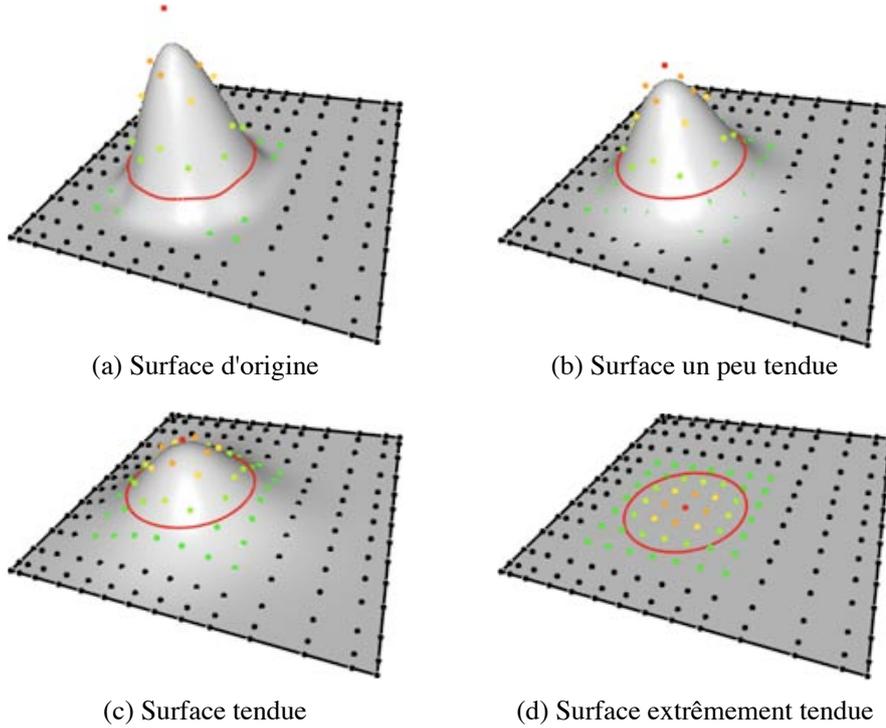


Figure 4.28. Aplanir, tendre ou lisser une zone

Cette méthode converge relativement lentement. L'objectif de la thèse n'étant pas pour l'instant la vitesse, l'étude de l'accélération de convergence de cette technique est une nouvelle perspective d'évolution. Dans le cas d'une surface dont le degré est égal à 1 dans les deux directions, cette technique ne peut pas fonctionner puisque la surface passe déjà par les points de contrôle.

4.3.4.5 Aplatir une zone

L'action d'*aplatir* une zone est plus forte qu'un simple lissage puisque la zone doit devenir entièrement plate, cette déformation ne nécessite pas de paramètre. Sa mise en œuvre nécessite le calcul du *plan moyen* \mathcal{P} de l'ensemble \bar{P} des pôles de la zone à déformer. Il suffit alors de projeter tous les pôles de \bar{P} sur \mathcal{P} pour aplatir la zone. Le résultat ne peut être plan que si le nombre de pôles impliqués dans la déformation est supérieur à $(p + 1 \times q + 1)$ c'est-à-dire 4×4 pôles pour une surface de degré 3×3 . C'est pourquoi le

seuil de raffinement de cette déformation contenu dans la base de connaissances est assez faible⁶ (10%). La figure 4.29 illustre ce traitement sur une courbe et la figure 4.30 donne un exemple appliqué à une zone sur une surface.

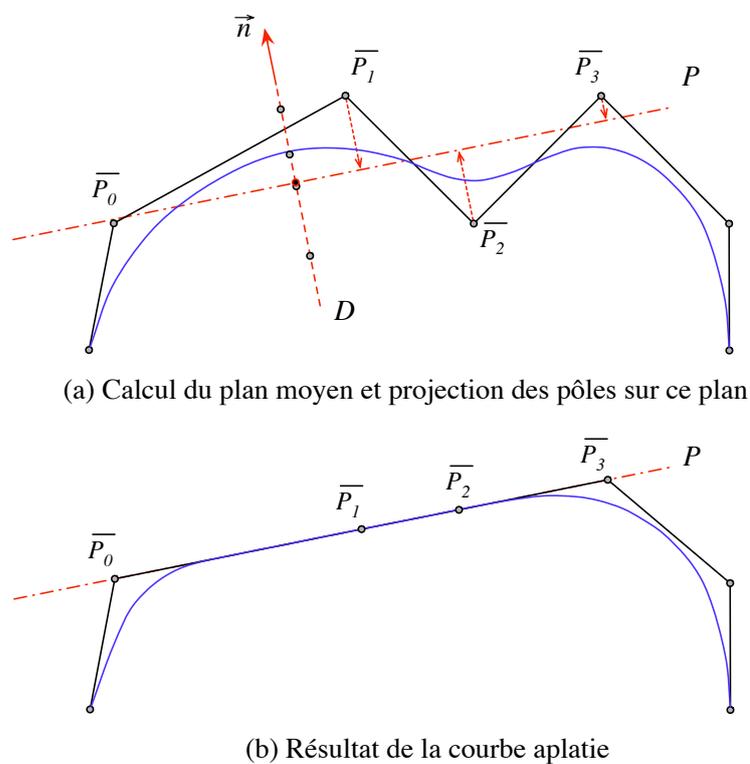


Figure 4.29. Principe du traitement «aplatir» appliqué sur une courbe

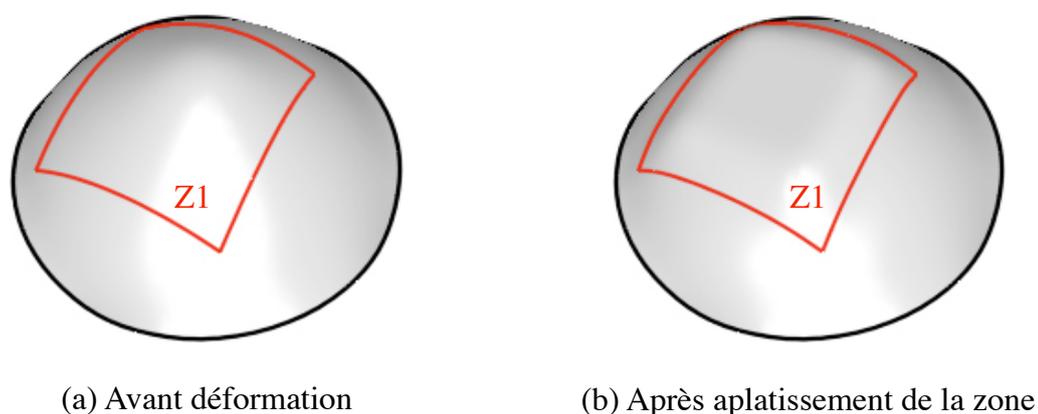


Figure 4.30. Aplatissement d'une zone

⁶nous rappelons qu'au plus le seuil est faible au plus la densité de pôle est élevée

4.3.4.6 Extruder ou Estamper / Bomber/Creuser une zone

Réaliser une *extrusion* ou une *estampe* sur une surface correspond à imprimer en relief ou en creux l’empreinte d’une forme. Nous traitons donc ces déformations comme des cas particuliers d’une déformation de type *bomber* ou *creuser*.

De façon générale, pour gérer une déformation de type *bombé*, nous choisissons d’appliquer à chacun des pôles associés à la zone considérée, un déplacement dans la direction de la normale moyenne de cette zone. Le déplacement associé à chaque pôle \bar{P}_i de \bar{P} voit sa distance varier en fonction de la valeur I_i correspondant à \bar{P}_i dans la grille de distances de la zone considérée. Nous considérons ainsi que le déplacement à apporter aux pôles est d’autant plus important que l’on s’éloigne des frontières de la zone considérée, c’est-à-dire que l’on se rapproche de son centre. Le déplacement à appliquer au pôle \bar{P}_i est le suivant :

$$\vec{d}_i = f_b(I_i) \mathcal{F}_w(Q) \vec{n} \quad (4.4)$$

La fonction f_b qui intervient dans la formule (4.4) a pour but de mieux contrôler la déformation de la zone en faisant intervenir un coefficient de bombé que l’on note $b \in \mathbb{R}^{+*}$. Cette fonction est définie telle que $f_b : \mathbb{R}^+ \mapsto \mathbb{R}$ et varie selon l’effet désiré :

- Les actions de *bomber* ou de *creuser* doivent permettre une certaine souplesse sur le contrôle de la déformation. Ainsi il faut pouvoir donner à l’utilisateur la possibilité de choisir entre plusieurs types de bombé ou de creusé. Nous définissons donc les fonctions de contrôle suivantes :

$$\text{Bomber : } f_b(x) = |x|^{1/b} \quad \text{Creuser : } f_b(x) = -|x|^{1/b} \quad (4.5)$$

La figure 4.31 présente la variation de cette fonction de contrôle :

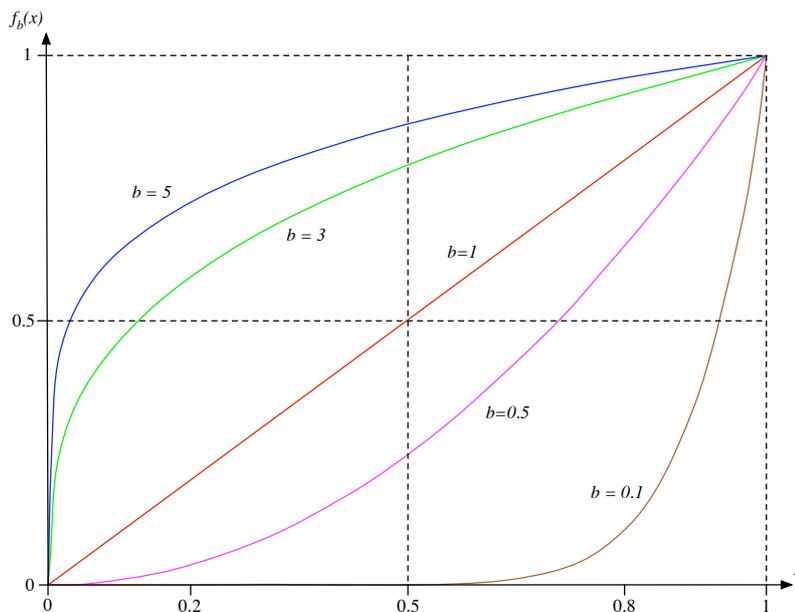


Figure 4.31. Variation de la fonction de contrôle selon le coefficient de bombement

La figure 4.32 schématise, sur une courbe, le procédé que nous venons de décrire et la figure 4.33 illustre son impact sur une surface initialement *peu bombée*.

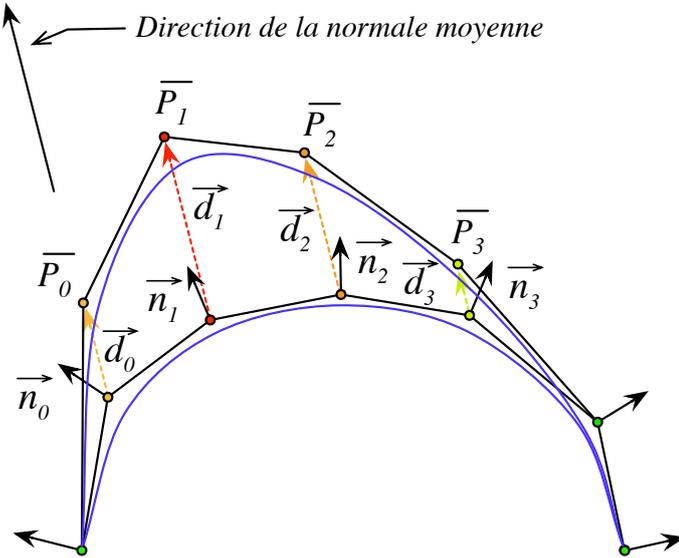


Figure 4.32. Principe du traitement «*bomber*» appliqué sur une courbe

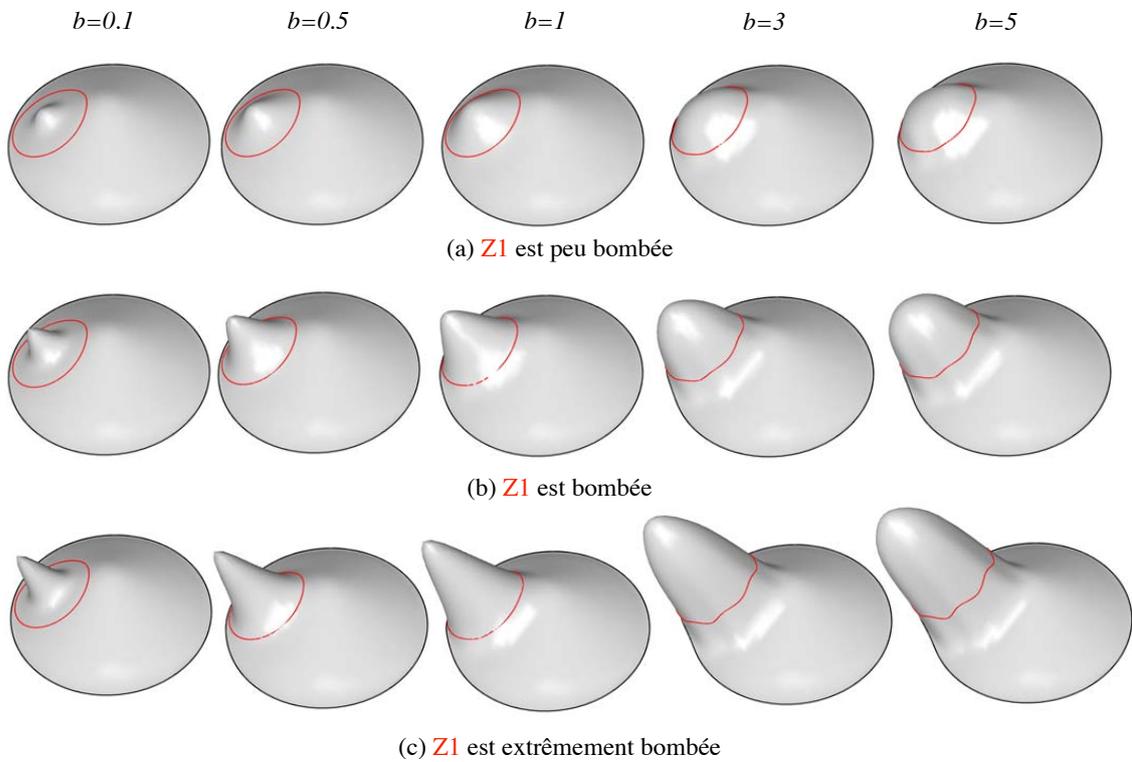


Figure 4.33. Bombement d'une zone en fonction du quantificateur (pour les lignes) et du coefficient de bombement b (pour les colonnes)

Pour une valeur de $b < 1$, la déformation se présente sous une forme très pointue, alors que si on a $b \geq 1$ la déformation est de plus en plus arrondie. Il est donc possible de jouer sur deux paramètres pour définir une déformation «bomber» : le paramètre b qui donne l'allure plus ou moins arrondie de la forme et le quantificateur général $\mathcal{F}_w(Q)$ qui contrôle l'amplitude de la déformation. Il est à noter, comme il est possible de le voir sur la figure 4.33, que les déformations engendrées par les valeurs de $b < 0.5$ peuvent difficilement être considérées comme bombées. Elles figurent simplement sur l'illustration pour mieux appréhender l'impact du paramètre b sur la déformation.

L'utilisateur n'a pas à donner explicitement ces valeurs, il peut cependant préciser le type de *bombé* qu'il désire en disant par exemple : «Z1 est peu bombée selon une forme très arrondie». Dans ce cas, l'utilisateur quantifie par *peu* ($\mathcal{F}_{w1}(Q1)$) l'amplitude de la déformation et par *très* ($\mathcal{F}_{w2}(Q2)$) l'aspect arrondi, c'est-à-dire b . Une heuristique donnant de bons résultats et permettant d'obtenir la valeur de b en fonction d'une quantification $\mathcal{F}_{w2}(Q2)$ est de dire que :

$$b = 0.5 + \left(\frac{\mathcal{F}_{w2}(Q2)}{50} \right)^2 \quad (4.6)$$

Ainsi si $\mathcal{F}_{w2}(Q2) \in [0, 100]$ alors $b \in [0.5, 4.5]$ avec une valeur moyenne lorsque le quantificateur $\mathcal{F}_{w2}(Q2) = 50$ égale à 1.5. Dans le cas où $\mathcal{F}_{w2}(Q2)$ est strictement supérieur à 100, ce qui peut être le cas lorsque l'utilisateur utilise un effet d'insistance, alors le degré augmente rend la forme du *bombé* encore plus arrondie.

Actuellement, si aucune information ne précise la forme du *bombé* dans la description, nous considérons simplement que $b = 1.5$. La quantification de la déformation «bombée» ne contrôle alors que l'amplitude. Cela fait apparaître un nouveau besoin à considérer en perspective qui correspond au concept de *configuration* de déformations. Comme nous l'avons déjà vu avec les formes, une forme peut avoir plusieurs configurations possibles. L'idée de la configuration des déformations se base sur le même principe. Ainsi pour la déformation *bombée*, si l'utilisateur ne précise pas la forme de la déformation, il faudrait que la complétion symbolique distingue plusieurs types de bombés en jouant sur la propriété *arrondie*. Ceci permettrait de proposer plusieurs types de bombés comme résultats possibles ;

- Les effets d'*extrusion* consistent à déplacer tous les pôles de la zone sans prendre en compte les valeurs définies dans la grille de distances de la zone. Ces cas sont les cas limites des actions de *bomber* et de *creuser* lorsque le coefficient de bombé tend vers l'infini. Ainsi la fonction f_b se résume à $f_b(x) = 1$ pour une extrusion en relief et $f_b(x) = -1$ pour une extrusion creusée. La figure 4.34 schématise, sur une courbe, le procédé d'extrusion que nous venons de décrire.

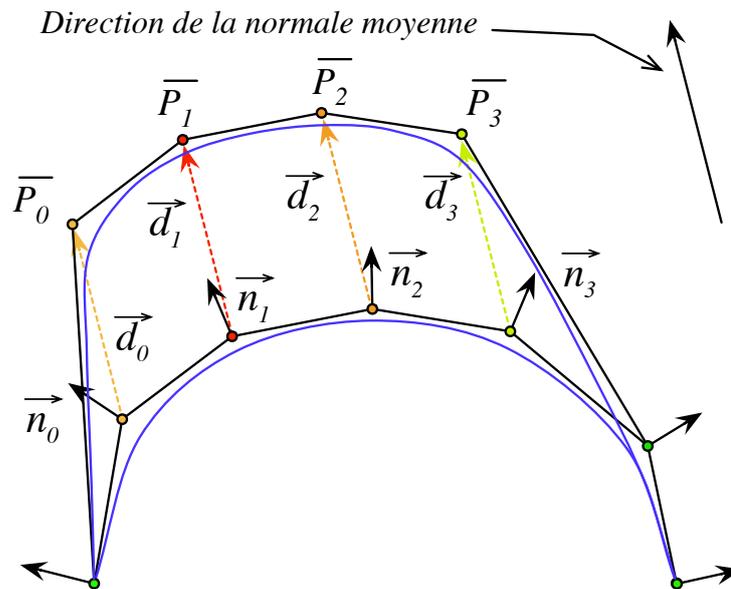


Figure 4.34. Principe du traitement «*extruder*» appliqué sur une courbe

La figure 4.35 illustre une extrusion appliquée sur une surface initialement *peu bombée* à partir d'une zone de forme quelconque. On peut d'ailleurs remarquer que les bords de la zone extrudée ne sont pas toujours très lisses et que la technique décrite voit son résultat très dépendant de la répartition des points de contrôle avant la déformation.

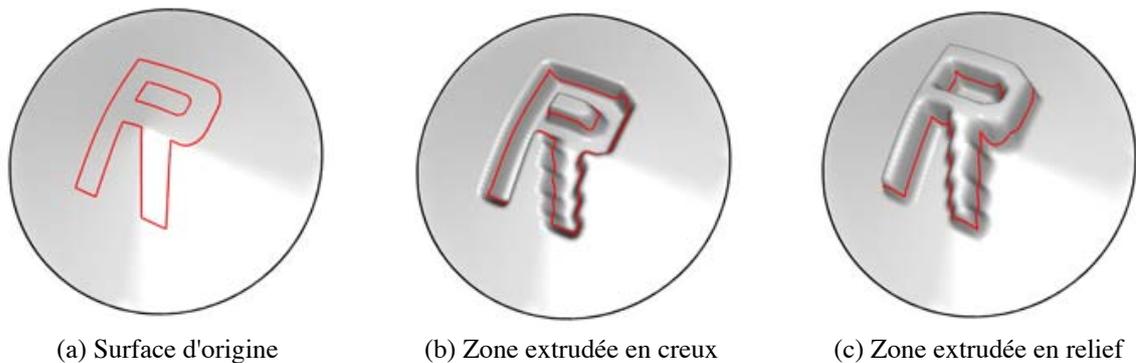


Figure 4.35. Extrusion d'une zone

4.3.4.7 Gonfler/Dégonfler une zone

À la différence d'une déformation «*bombée*», le gonflement n'utilise pas seulement la normale moyenne de la zone, mais plutôt chaque normale des pôles impliqués dans la déformation. L'avantage d'avoir une direction différente pour chaque pôle permet un déplacement qui n'est pas globalement rectiligne comme nous venons de le voir dans la manipulation précédente.

L'heuristique que nous proposons pour gérer cette déformation permettant de *gonfler* une

zone est itérative. Elle se base sur les normales de chaque pôle et sur la grille de distances de la zone à déformer. Ainsi à chaque itération, nous appliquons à chaque pôle \bar{P}_i de \bar{P} attaché à la zone, un déplacement dans la direction de sa normale \vec{n}_i d'une distance égale à la valeur qui lui est associée dans la grille de distances I_i pondérée par le quantificateur de la déformation $\mathcal{F}_w(Q)$. Le déplacement \vec{d}_i à appliquer lors d'une itération au pôle \bar{P}_i se formalise comme suit :

$$\vec{d}_i = I_i \frac{\mathcal{F}_w(Q)}{50^2} \vec{n}_i \quad (4.7)$$

Nous choisissons cette formulation car I_i a ses valeurs qui partent de 0 aux bords de la zone et qui se rapprochent d'autant plus de 1 que l'on s'approche du *noyau*⁷ de cette même zone. Ce qui permet d'appliquer des déplacements plus faibles lorsque l'on est sur les bords et plus grands au centre. Ce comportement a alors tendance à attirer vers le haut les pôles du centre de la zone et à incliner les normales à la périphérie en déplaçant peu les pôles qui y sont rattachés. Cette opération est réalisée itérativement de manière à incliner de plus en plus, à chaque itération, les normales périphériques à la zone permettant ainsi d'élargir la déformation au lieu de simplement tirer la surface vers le haut. La pondération par $\frac{1}{50^2}$ permet de limiter la distance le déplacement et donc d'accentuer ce phénomène d'inclinaison des normales aux pôles. Si le nombre d'itérations est trop élevé, il peut apparaître des effets d'auto-intersections sur les bords dûs à l'enroulement de la surface sur elle-même. Il faut donc limiter ce nombre d'itérations afin d'avoir une déformation visuellement correcte. L'heuristique choisie est très simple, elle consiste à appliquer un nombre d'itérations égal à $\mathcal{F}_w(Q)$.

Les résultats obtenus correspondent à notre attente et permettent d'obtenir l'effet de gonflement recherché. la figure 4.36 illustre ce traitement sur une courbe et la figure 4.37 donne un exemple appliqué à une zone sur une surface.

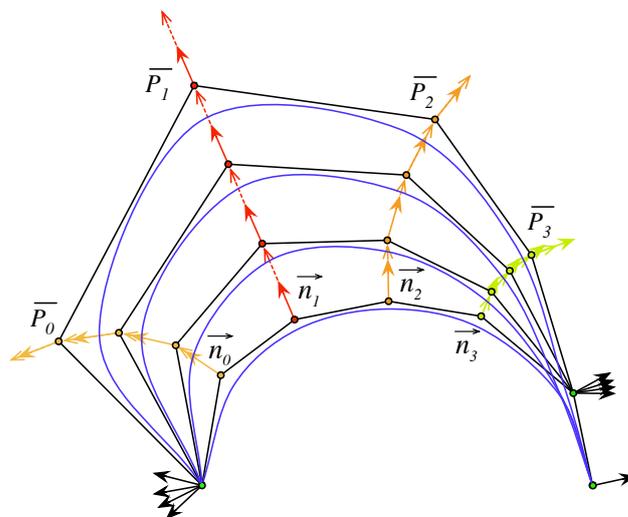


Figure 4.36. Principe du traitement «*gonfler*» appliqué sur une courbe

⁷partie considéré comme la plus éloignée des frontières de la zone

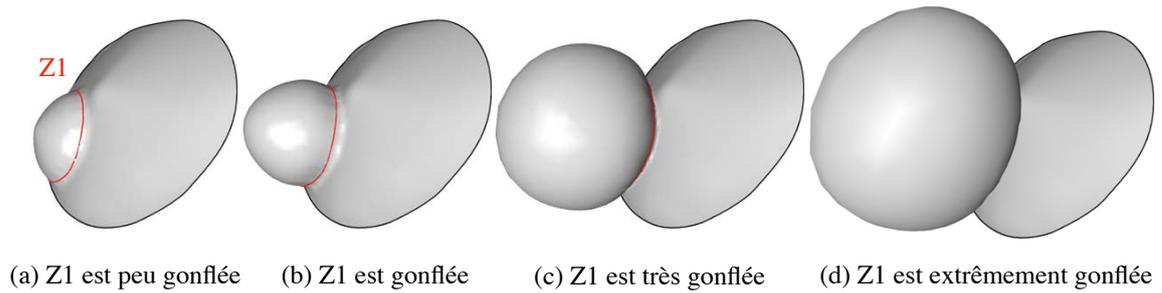


Figure 4.37. Gonflement d'une zone

L'action de *dégonfler* est l'opération opposée à l'action de *gonfler*, le déplacement appliqué à chaque pôle est donc cette fois-ci dans la direction opposée à la normale. Il est important de noter qu'un dégonflement ne peut pas donner le résultat escompté sur une zone n'ayant par été gonflée auparavant.

Ces deux traitements nécessitent un nombre de pôles suffisant pour obtenir le résultat recherché. De plus, dans le cas d'une surface très déformée où le polyèdre de contrôle admet des parties concaves, les normales aux pôles ont des directions qui peuvent être très différentes voire opposées ce qui peut entraîner l'apparition d'auto-intersections. Un moyen d'obtenir des résultats plus satisfaisants est, de façon transparente pour l'utilisateur, d'aplatir la zone avant d'appliquer une déformation de *gonfler*.

4.3.4.8 Trouer une zone

Bien que ce ne soit pas vraiment une déformation, puisqu'il n'y a aucun déplacement de point de contrôle, nous avons choisi de faire apparaître cette action dans ce paragraphe puisqu'elle fait partie des outils de construction du modèle géométrique. Faire un trou dans une surface correspond simplement à réaliser une restriction dans le domaine paramétrique de cette surface comme cela a déjà été présenté dans la partie 1.3.4.1 de ce document. Une zone étant, dans notre système, déjà définie dans le plan paramétrique de la surface, il suffit de dire que cette zone est une restriction du domaine paramétrique. La figure 4.38 illustre ce concept.

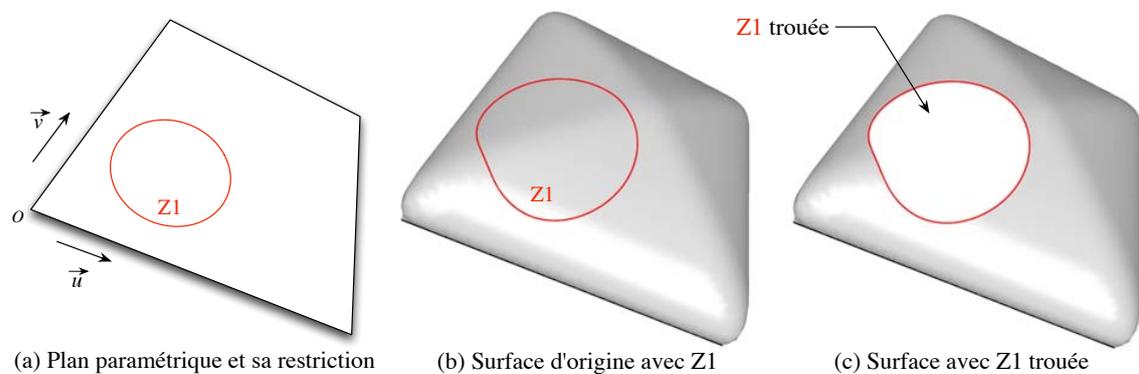


Figure 4.38. Trouer une zone de la surface

4.3.4.9 Bosseler une zone

Bosseler une zone de la surface consiste à faire apparaître un ensemble de *vagues* constituées de creux et de bosses. Ces perturbations doivent cependant rester d'une amplitude relativement faible pour ne pas complètement perdre la forme existante de la surface. Pour cela nous proposons de déplacer les pôles de la zone de façon alternée, une fois dans la direction de la normale du pôle et une fois dans la direction opposée à la normale du pôle. La distance du déplacement dépend du quantificateur contenu dans le modèle sémantique généralisé et peut être perturbée par une fonction $H(\xi)$. $H(\xi)$, que l'on nomme fonction de perturbation, est définie telle que : $H(\xi) \mapsto [1 - \xi, 1]$ avec $\xi \in [0, 1]$ considéré comme le coefficient de perturbation à appliquer. Un appel à $H(\xi)$ consiste à tirer aléatoirement une valeur dans l'intervalle $[1 - \xi, 1]$. Ainsi $H(\xi)$ change de valeur à chacun de ces appels et permet de briser la régularité de la méthode.

La formulation du déplacement à appliquer à chaque pôle \bar{P}_i de \bar{P} où \bar{P}_i peut aussi être repéré sur la grille de contrôle par les coordonnées (k, l) , est alors la suivante :

$$\vec{d}_i = \begin{cases} H(\xi) \mathcal{F}_w(Q) \vec{n}_i & \text{Si } k + l \text{ est pair} \\ -H(\xi) \mathcal{F}_w(Q) \vec{n}_i & \text{Sinon} \end{cases} \quad (4.8)$$

Cette déformation est encore contrôlée par deux paramètres, ξ qui correspond à sa perturbation et $\mathcal{F}_w(Q)$ qui correspond ici à son amplitude. L'importance de la notion de configuration des déformations est donc encore présente. Ce concept n'étant pas encore intégré à notre processus, nous utilisons actuellement une heuristique qui consiste à dire que l'utilisateur quantifie implicitement la notion d'amplitude et de perturbation dans le terme de *bosselé*. On considère dans ce cas ξ comme étant égale à $\frac{\mathcal{F}_w(Q)}{100}$ majorée par 1.

La figure 4.39 schématise la méthode sur une courbe et la figure 4.40 l'illustre sur une surface.

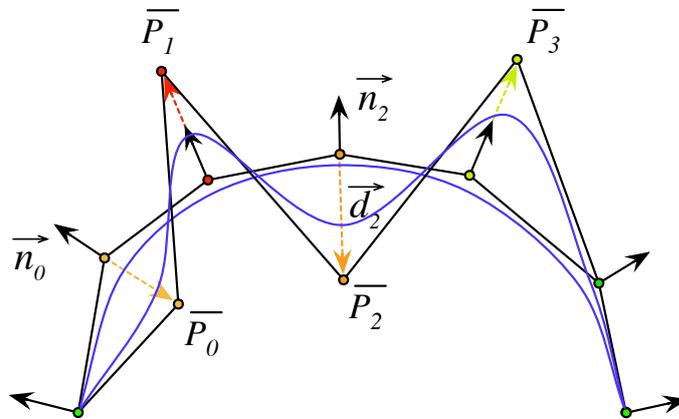


Figure 4.39. Principe du traitement «bosseler» (avec perturbation) appliqué à une courbe

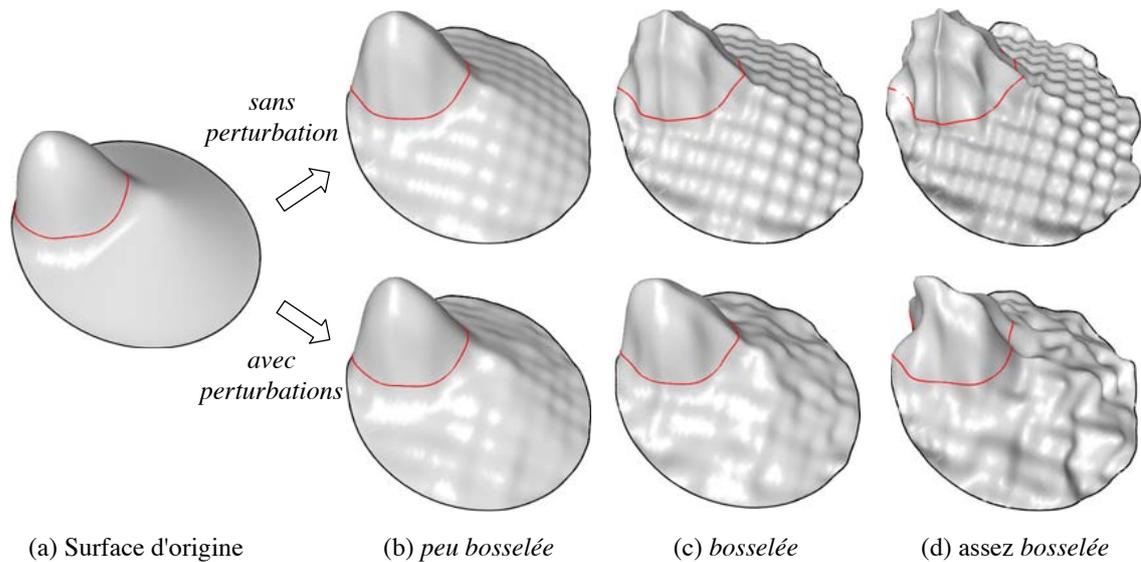


Figure 4.40. Bosseler une zone

4.3.4.10 Courber, plier, corner ou enrouler une zone

La déformation *courber* consiste à légèrement replier la surface sur elle-même. On pourrait comparer cette déformation à l'action consistant à plus ou moins enrouler la surface autour d'un cylindre. Au plus l'utilisateur insiste sur le fait que la surface est courbée au plus l'enroulement autour du cylindre est important. Ainsi, pour contrôler cette déformation, nous utilisons un paramètre α . Celui-ci est l'angle total à appliquer pour simuler cet enroulement. C'est lui qui varie en fonction de la quantification de la déformation précisée par l'utilisateur. Il est ainsi possible d'obtenir des résultats plus ou moins cornés ou courbés selon la demande de l'utilisateur. Par défaut, un quantificateur moyen à 50% correspond à un angle $\alpha = 90^\circ$. Si cet angle est positif, la déformation se fera vers le haut⁸ (direction de la normale) et s'il est négatif la déformation sera réalisée vers le bas.

Cette déformation n'est envisageable que s'il est possible de définir un *axe paramétrique de courbure* qui correspond à une ligne du plan paramétrique à partir de laquelle on peut commencer à courber la surface. Ceci contraint fortement la forme et la position d'une zone pouvant être courbée. Celle-ci doit forcément avoir ses frontières confondues avec les bords du plan paramétrique, excepté un unique segment qui doit relier deux bords différents du plan paramétrique. Nous considérons cet unique segment comme étant l'*axe paramétrique de courbure*. La figure 4.41 présente plusieurs cas possibles de définition de cet axe dans le plan paramétrique alors que la figure 4.42 illustre des cas ne permettant pas de le déterminer puisqu'aucun segment ne vérifie la condition précédemment exprimée.

⁸relativement au référentiel défini à la partie 4.3.3, étape 1

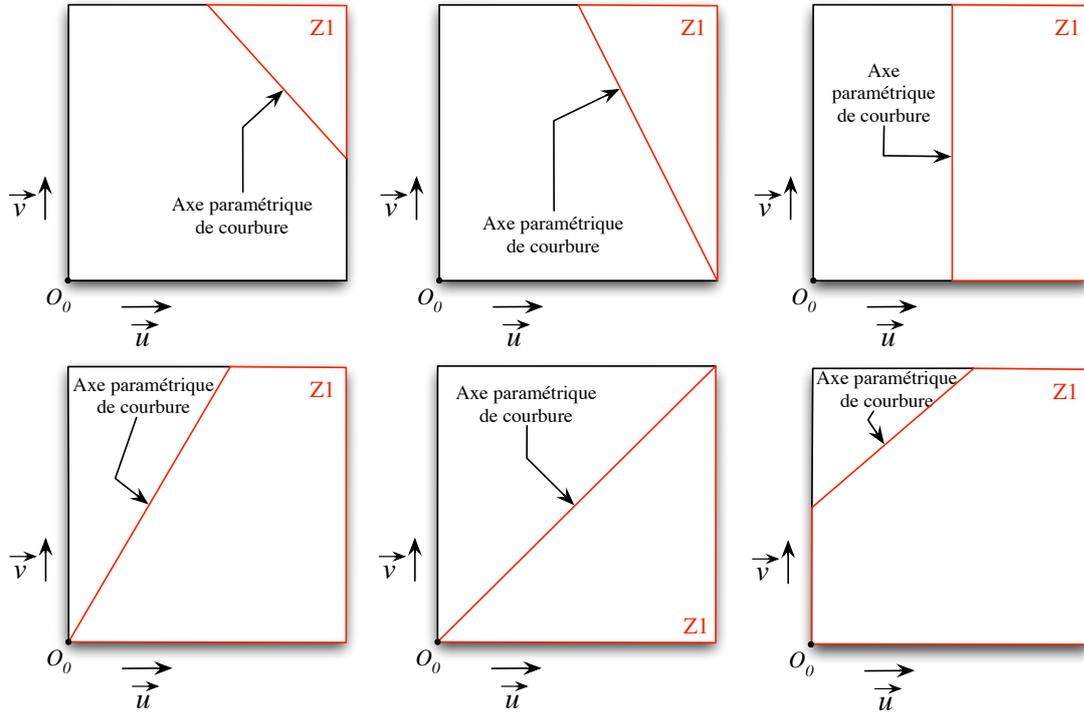


Figure 4.41. Exemples de zones pouvant être courbées, accompagnées de leur axe paramétrique de courbure

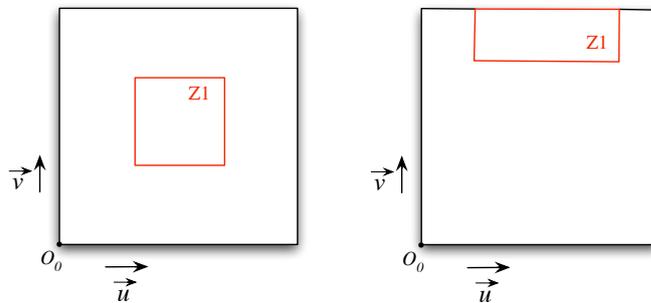


Figure 4.42. Exemples de zones ne pouvant pas être courbées

D'un point de vue paramétrique, nous connaissons l'axe à partir duquel il faut commencer à courber la surface. Cet axe paramétrique de courbure est noté $A_p(1)$. Nous connaissons aussi les pôles \bar{P}_i impliqués dans la déformation. Ce sont ceux qui sont rattachés à la zone à déformer. L'heuristique que nous proposons commence par repérer l'ensemble des images paramétriques de ces pôles notées IP_i par rapport à $A_p(1)$ dans le plan paramétrique. Pour cela, nous choisissons d'associer à chaque image paramétrique IP_i la distance qui la sépare de $A_p(1)$, c'est-à-dire la distance euclidienne qui la sépare de sa projection orthogonale sur $A_p(1)$. La figure 4.43 présente cette projection orthogonale sur un exemple de zone correspondant à la partie inférieure à la diagonale Bas-Gauche, Haut-Droite du plan paramétrique.

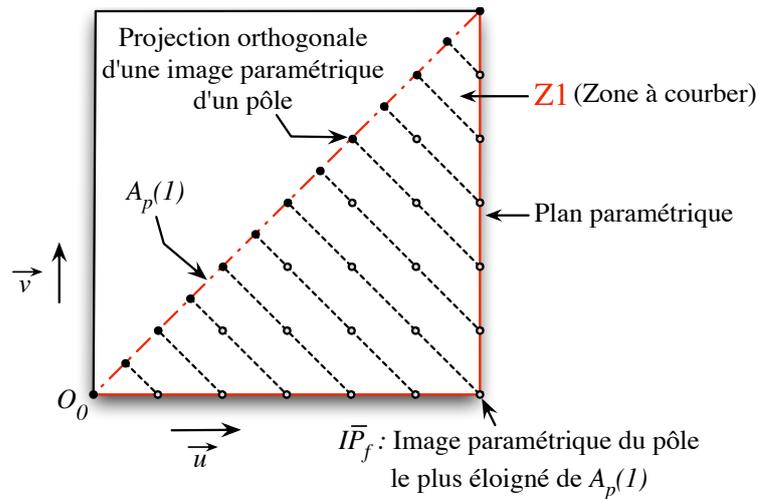


Figure 4.43. Projection des images paramétriques des pôles sur $A_p(1)$

Nous définissons alors un paramètre C_e qui va nous permettre de regrouper en intervalles les images paramétriques des pôles en fonction de leur distance à $A_p(1)$. Soit C_e le nombre d'intervalles à considérer. Tous les intervalles sont définis comme ayant la même taille égale à la distance de l'image du pôle le plus éloigné de $A_p(1)$ divisé par C_e . Ainsi si $C_e = 1$, il n'existe qu'un seul intervalle regroupant toutes les images paramétriques des pôles. Si par contre il existe plusieurs intervalles, la zone paramétrique à déformer est découpée en bandes régulières parallèles à $A_p(1)$ et chaque intervalle contient alors une parties des pôles à déplacer. La figure 4.44 reprend la figure 4.43 en illustrant les intervalles correspondants à une valeur $C_e = 3$.

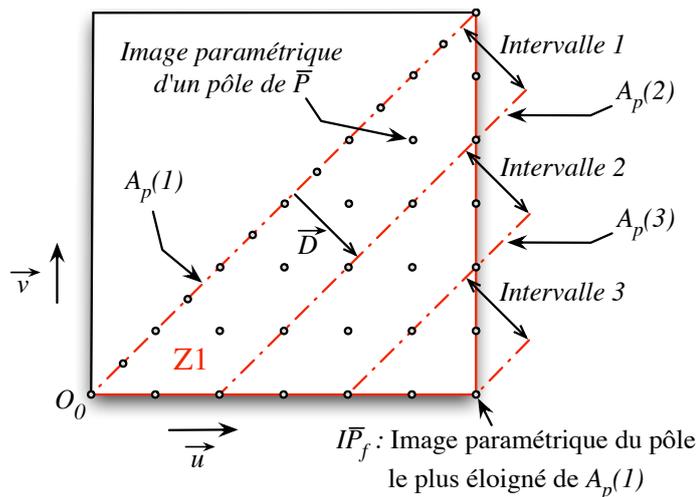


Figure 4.44. Définition des intervalles de travail pour $C_e=3$

La déformation désirée consiste à enrouler ou à courber la surface. Notre approche consiste à associer chaque intervalle à une rotation particulière. Chaque intervalle est donc rattaché à un angle $\frac{\alpha}{C_e}$ afin de répartir régulièrement la rotation à réaliser. Il est aussi lié à un axe

paramétrique de rotation. Cet axe $A_p(k)$ local à chaque intervalle k , n'est autre que la droite qui détermine le début de l'intervalle k dans le plan paramétrique. Ainsi, le premier intervalle a pour axe paramétrique de rotation $A_p(1)$. La figure 4.44 présente une illustration des axes paramétriques attachés aux intervalles.

Toutes les données paramétriques nécessaires au traitement sont définies. Il nous faut maintenant passer au niveau géométrique pour appliquer de façon effective la déformation en agissant sur les points de contrôle de la surface. Pour cela, nous souhaitons appliquer à chaque pôle une rotation dépendante de l'intervalle auquel il appartient. Nous avons ainsi besoin d'un axe de rotation dans l'espace cartésien. Cet axe est le représentant de $A_p(k)$ dans cet espace. Nous le nommons $A_g(k)$ pour *axe géométrique de courbure*. Nous le déterminons par une heuristique qui consiste à trouver la droite passant au plus près des pôles dont l'image paramétrique est proche de $A_p(k)$.

La technique que nous utilisons pour trouver l'axe géométrique $A_g(k)$ correspondant à un axe paramétrique $A_p(k)$ amène à l'introduction de la notion de *pôles influents*. L'axe paramétrique $A_p(k)$ traverse un ensemble de pavés paramétriques $[U_i, U_{i+1}] \times [V_j, V_{j+1}]$. Nous appelons *pôles influents* sur l'axe $A_p(k)$ tous les points de contrôle intervenant dans la définition de la surface sur ces pavés. La première étape de cette technique consiste à déterminer le plan moyen \mathcal{P} (Cf. 4.3.4.3) des points de contrôle influents sur $A_p(k)$. Nous considérons que $A_g(k)$ doit appartenir à ce plan. Dans la deuxième étape, nous projetons sur \mathcal{P} ces mêmes pôles. L'axe $A_g(k)$ est alors la droite de \mathcal{P} passant au plus près de ces pôles projetés. Chaque pôle est pondéré par son poids dans la définition de la surface NURBS ainsi que par la distance qui sépare son image paramétrique de l'axe $A_p(k)$.

Remarque 4.3.2 – *Dans la version actuelle de nos travaux, nous utilisons des surfaces NURBS dont tous les poids sont égaux à 1. Ces surfaces ne sont donc, pour le moment, que des B-splines. Une perspective d'évolution des travaux est d'exploiter les poids de chaque pôle afin de profiter pleinement des capacités des NURBS.*

Tous les éléments sont réunis pour appliquer la technique complète. Celle-ci consiste à traiter chaque intervalle les uns après les autres. Le traitement d'un intervalle k correspond à l'application d'une rotation d'axe $A_g(k)$ et d'angle $\frac{\alpha}{C_e}$ à tous les pôles \bar{P}_i contenus dans tous les intervalles à partir de l'intervalle courant k , c'est-à-dire tous les intervalles l avec $l \in [k, C_e]$. Une fois l'intervalle traité, il faut passer à l'intervalle suivant ($k = k + 1$). Celui-ci ayant son propre $A_p(k)$, l' $A_g(k)$ correspondant doit être déterminé avant d'appliquer à nouveau les rotations sur les intervalles restants.

Exemple : Cet exemple est appliqué à une courbe pour une meilleure visibilité, mais peut être assimilé à la coupe d'une surface initialement plane. Dans cet exemple nous considérons $C_e = 3$ et $\alpha = 120^\circ$. Nous connaissons initialement $A_p(1)$ qui est déterminé par la définition de la zone, ainsi que l'ensemble \bar{P} des pôles impliqués dans la déformation et leurs images paramétriques $I\bar{P}$. À partir de ces informations, nous calculons les intervalles et l'axe géométrique correspondant à $A_p(1)$. La figure 4.45 présente ces conditions initiales.

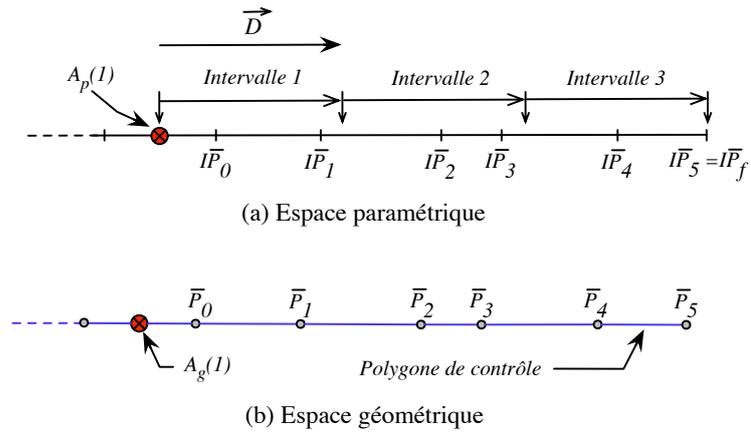


Figure 4.45. Conditions initiales d'une déformation «courber»

Traitement de l'intervalle 1 : Nous appliquons à tous les pôles dont les images sont dans les intervalles 1, 2 et 3, une rotation d'axe $A_g(1)$ et d'angle $\frac{\alpha}{C_e} = 40^\circ$. Cela conduit au résultat de la figure 4.46.

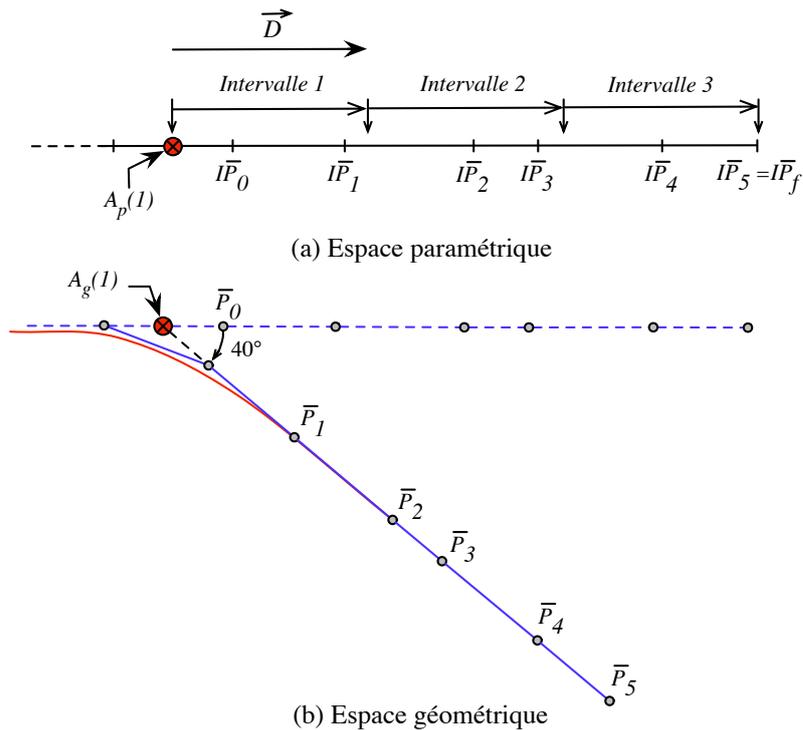


Figure 4.46. Ex. de déformation «courber» : traitement de l'intervalle 1

Traitement de l'intervalle 2 : $A_p(2)$ est calculé tel que $A_p(2) = A_p(1) + \vec{D}$. Son correspondant $A_g(2)$ est déterminé dans l'espace géométrique et on applique à tous les pôles, dont les images appartiennent aux intervalles 2 et 3, une rotation d'axe $A_g(2)$ et d'angle 40° . Nous obtenons le résultat de la figure 4.47.

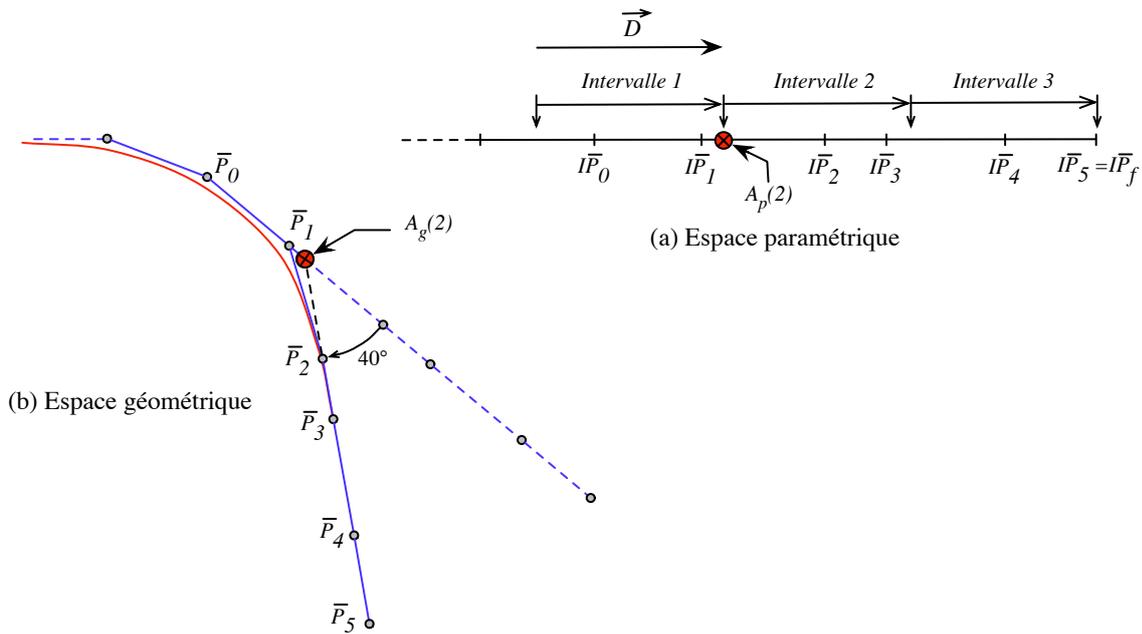


Figure 4.47. Ex. de déformation «courber» : traitement de l'intervalle 2

Traitement de l'intervalle 3 : $A_p(3)$ est calculé tel que $A_p(3) = A_p(1) + 2\vec{D}$. Son correspondant $A_g(3)$ est déterminé dans l'espace géométrique et on applique à tous les pôles, dont les images appartiennent à l'intervalle 3, une rotation d'axe $A_g(3)$ et d'angle 40° . Cela conduit au résultat de la figure 4.48.

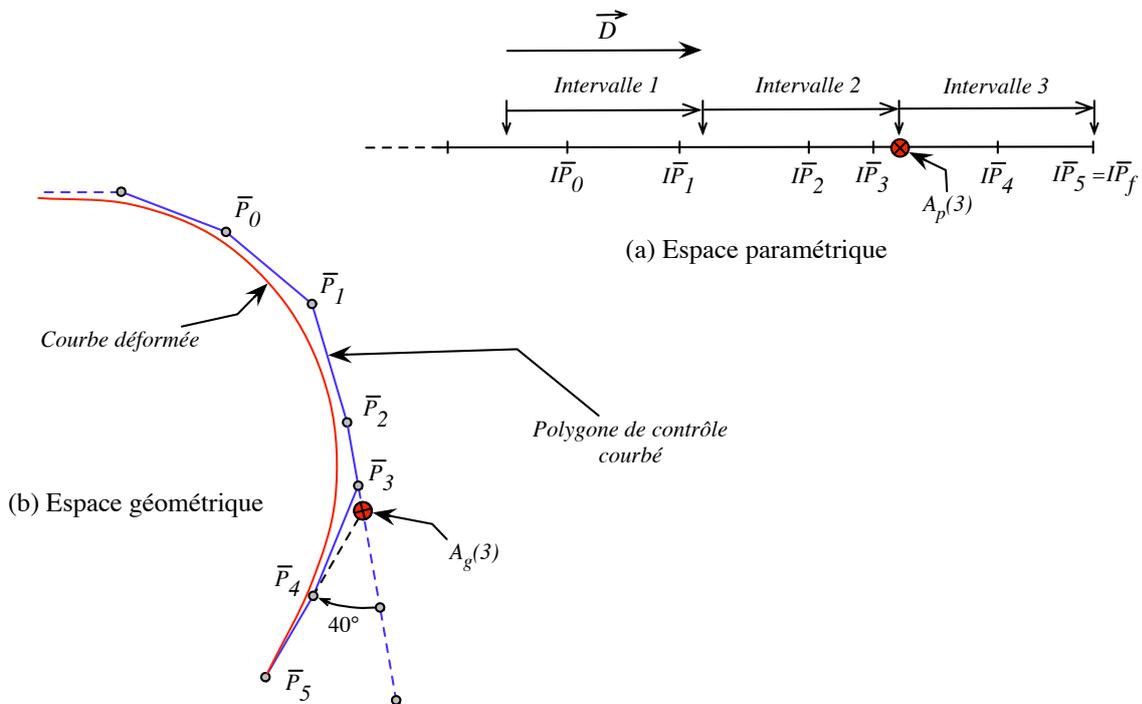


Figure 4.48. Ex. de déformation «courber» : traitement de l'intervalle 3

Il n'y a plus d'intervalle à traiter, la déformation est terminée. La figure 4.49 présente quelques résultats obtenus sur une surface initialement *peu bombée*.

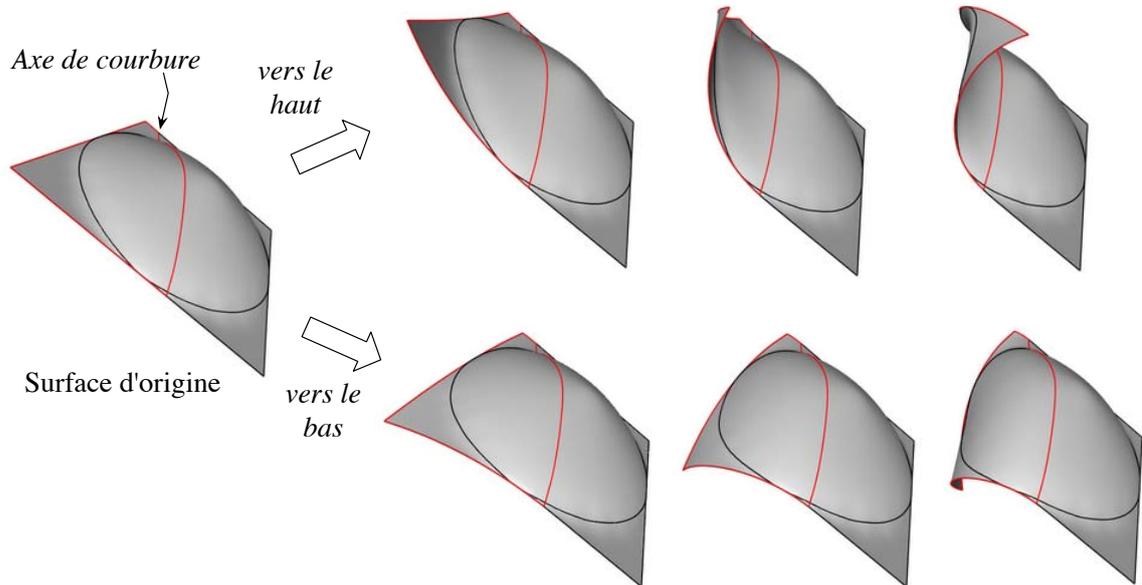


Figure 4.49. Courber une zone

Le choix de la valeur de C_e conditionne la forme arrondie de la déformation. Celle-ci est en effet plus régulière et plus arrondie si C_e est choisi grand⁹.

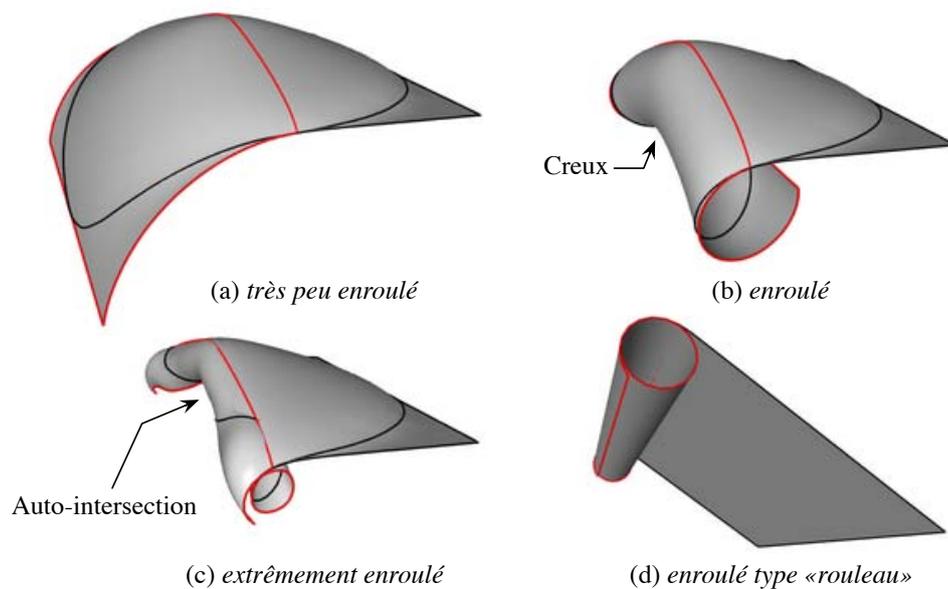


Figure 4.50. Enroulement d'une zone

⁹Une valeur considérée comme grande est par exemple le nombre de pôles de \bar{P}

Si l'on travaille avec des valeurs d'angle relativement élevées, la déformation initialement nommée *courber* ou *corner* devient un enroulement de la surface. Il est cependant important de noter que ce type de déformation est très dépendant de la forme initiale de la surface. Il peut donc apparaître des creux et même des auto-intersections si les valeurs de quantification utilisées sont trop grandes. Les figures 4.50.a, 4.50.b et 4.50.c présentent trois niveaux d'enroulement d'une même surface initialement *peu bombée*. La figure 4.50.d illustre, quant à elle, l'enroulement d'une surface plane que l'on peut associer à un *rouleau* ou à un *parchemin*.

Comme nous l'avons dit, le coefficient C_e correspond au nombre d'intervalles à considérer durant la déformation. Ainsi s'il vaut 1, il permet d'appliquer une simple rotation de tous les pôles de l'intervalle selon l'axe de courbure géométrique. Ce phénomène permet de réaliser des déformations de type «plier» que l'on illustre sur la figure 4.51. De plus, remarquons que la marque de pliure est d'autant plus marquée que le raffinement autour de l'axe de pliure est fin.

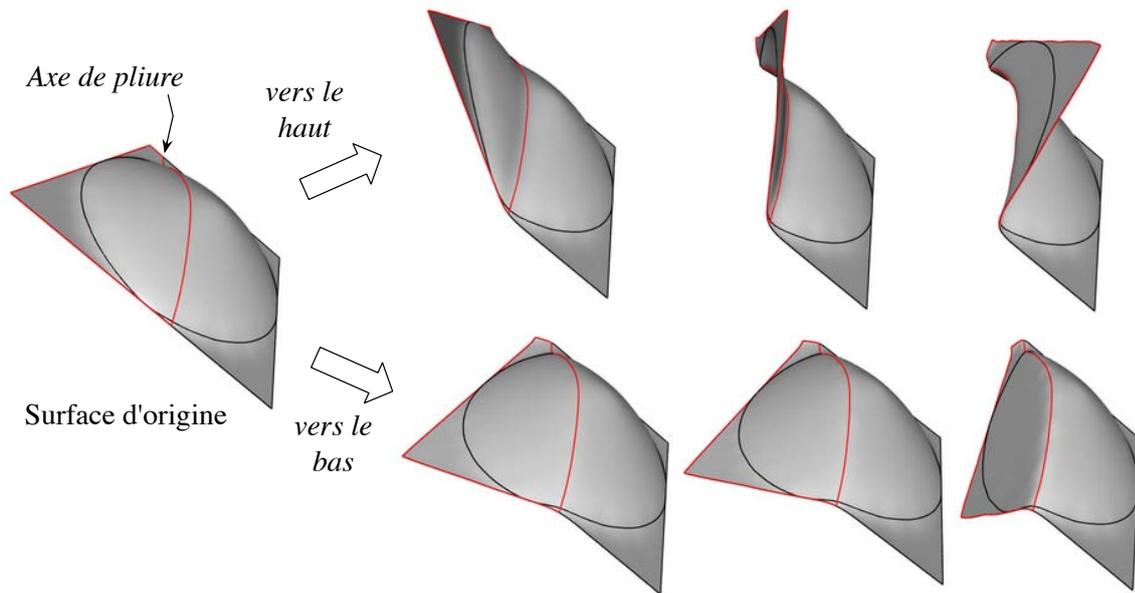


Figure 4.51. Plier une zone

De nombreuses autres déformations, ou contraintes «lâches», peuvent encore être traitées comme par exemple *tordre*, *vrier* ou même *plisser* une zone de la surface. Ces extensions n'ont pas été traitées dans le cadre de cette thèse et font l'objet des perspectives d'évolution du modelleur déclaratif de surfaces. Les parties suivantes de ce chapitre se penchent alors sur les contraintes dites «dures» que sont les contraintes de passage de la surface par un point particulier de l'espace. Nous commençons par résoudre ce problème pour une contrainte ponctuelle avec contrôle de son influence sur la surface, puis nous généralisons la méthode à N contraintes.

4.3.5 Satisfaction d'une contrainte ponctuelle avec contrôle d'influence

En CAO, la satisfaction de contraintes de passage est essentielle pour la réalisation de surfaces précises devant tenir compte d'autres objets ou pièces de son environnement. Ainsi il est possible de faire appel à des contraintes de passage pour le raccord entre plusieurs surfaces par exemple. Nous proposons dans cette partie une méthode permettant de parfaitement satisfaire une contrainte ponctuelle en contrôlant son influence [La Greca and al., 2004, La Greca and al., 2005]. Cette technique est d'abord proposée pour les courbes puis pour les surfaces.

4.3.5.1 Application aux courbes NURBS

En s'appuyant sur les définitions et les notations de la partie 1.3.4.2, une courbe NURBS \mathcal{C} de degré p s'écrit :

$$\mathcal{C}(u) = \sum_{i=0}^{n-1} R_i(u)P_i \quad (4.9)$$

où P_i est le point de contrôle i de poids w_i et où $R_i(u)$ s'écrit selon l'expression suivante :

$$R_i(u) = \frac{N_{i,p}(u)w_i}{\sum_{k=0}^{n-1} N_{k,p}(u)w_k}$$

Soit M un point de \mathcal{C} tel que $M = \mathcal{C}(u)$. Soit \hat{M} un point de l'espace, l'objectif de la méthode est de déformer la courbe \mathcal{C} pour obtenir la courbe $\hat{\mathcal{C}}$ telle que $\hat{\mathcal{C}}(u) = \hat{M}$. Le choix du paramètre u est traité dans le paragraphe 4.3.5.1.1.

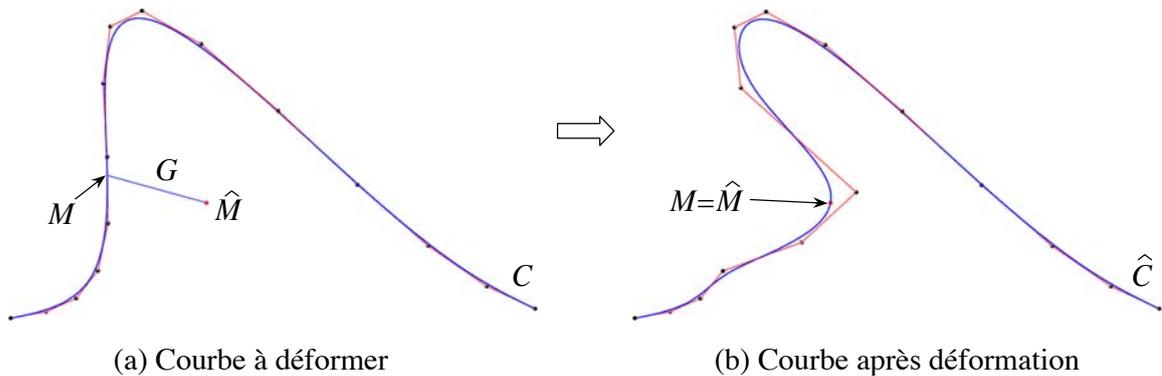


Figure 4.52. Exemple de contrainte de passage sur une courbe de degré 3, définie par 15 points de contrôle

On appelle alors G (Fig. 4.52) une contrainte qui doit vérifier un ensemble de trois conditions :

1. *contrainte spatiale* : la courbe \mathcal{C} doit passer par un point \hat{M} de l'espace ;
2. *contrainte paramétrique* : le point de passage de \mathcal{C} en \hat{M} doit se faire pour un paramètre spécifique u ;

3. *contrainte de localité* : chaque contrainte a une influence spécifique sur \mathcal{C} et ses points de contrôle. Cette influence se définit par la fonction $f(i)$ telle que :

$$\begin{cases} f : [0, n-1] \mapsto \mathbb{R}^+ \\ \exists i \in [0, n-1] / R_i(u)f(i) \neq 0 \end{cases}$$

Proposition 4.3.1 : Soit $\vec{e} = \overrightarrow{M\hat{M}}$, le vecteur déplacement $\overrightarrow{m(i)}$ à appliquer au point de contrôle P_i pour satisfaire la contrainte \mathcal{G} s'écrit de la manière suivante :

$$\overrightarrow{m(i)} = \frac{f(i)}{\sum_{k=0}^{n-1} [R_k(u)f(k)]} \vec{e} \quad (4.10)$$

Preuve :

Soit O l'origine du repère.

$$\begin{aligned} \overrightarrow{O\hat{\mathcal{C}}(u)} &= \sum_{i=0}^{n-1} [R_i(u)\overrightarrow{OP_i}] && \text{avec } \overrightarrow{OP_i} = \overrightarrow{OP_i} + \overrightarrow{m(i)} \\ &= \sum_{i=0}^{n-1} [R_i(u)(\overrightarrow{OP_i} + \overrightarrow{m(i)})] \\ &= \sum_{i=0}^{n-1} [R_i(u)\overrightarrow{OP_i}] + \sum_{i=0}^{n-1} [R_i(u)\overrightarrow{m(i)}] \\ &= \overrightarrow{OC(u)} + \sum_{i=0}^{n-1} [R_i(u)\overrightarrow{m(i)}] \end{aligned} \quad (4.11)$$

En utilisant (4.10) on obtient immédiatement :

$$\overrightarrow{O\hat{\mathcal{C}}(u)} = \overrightarrow{OC(u)} + \vec{e} = \overrightarrow{O\hat{M}} \quad (4.12)$$

□

Remarque 4.3.3 – $f(i)$ est définie sur \mathbb{R}^+ afin de garantir que le dénominateur de (4.10) soit strictement positif (avec des poids positifs ou nuls attachés aux points de contrôle). Toute fonction $f(i)$ est cependant valide sous réserve qu'elle vérifie :

$$\sum_{k=0}^{n-1} [R_k(u)f(k)] \neq 0$$

La figure 4.53.b illustre cette remarque en utilisant la fonction $f(i) = \cos(i)$.

4.3.5.1.1 Détermination du paramètre u

La contrainte \mathcal{G} a pour effet de déformer \mathcal{C} afin d'obtenir $\hat{\mathcal{C}}$ passant par \hat{M} en un point de paramètre u . Deux possibilités sont à considérer : soit u est fixé par l'utilisateur et la contrainte est *rigide*, soit aucune information n'est précisée sur u et le système doit trouver le paramètre le plus adapté. Dans ce dernier cas, nous considérons que la partie de la courbe la plus propice à être déformée est celle qui est la plus proche de \hat{M} . Ainsi, le paramètre u choisi est celui qui minimise la distance euclidienne entre $\mathcal{C}(u)$ et \hat{M} . Ce type de problème est difficile à résoudre à cause de la présence possible de minima locaux. Plusieurs heuristiques peuvent être utilisées [Moron, 1996, Dyllong and Luther, 1999], mais s'il existe plus d'une solution, seul l'utilisateur peut enlever l'ambiguïté.

4.3.5.1.2 Influence de la fonction $f(i)$

La fonction $f(i)$ contrôle l'influence de G sur l'ensemble des points de contrôle de \mathcal{C} . Ainsi, pour une même *contrainte spatiale* (Fig. 4.53.a) la courbe \mathcal{C} peut être déformée de différentes manières (Fig. 4.53.b, 4.53.c, 4.53.d) en fonction de sa *contrainte de localité* $f(i)$.

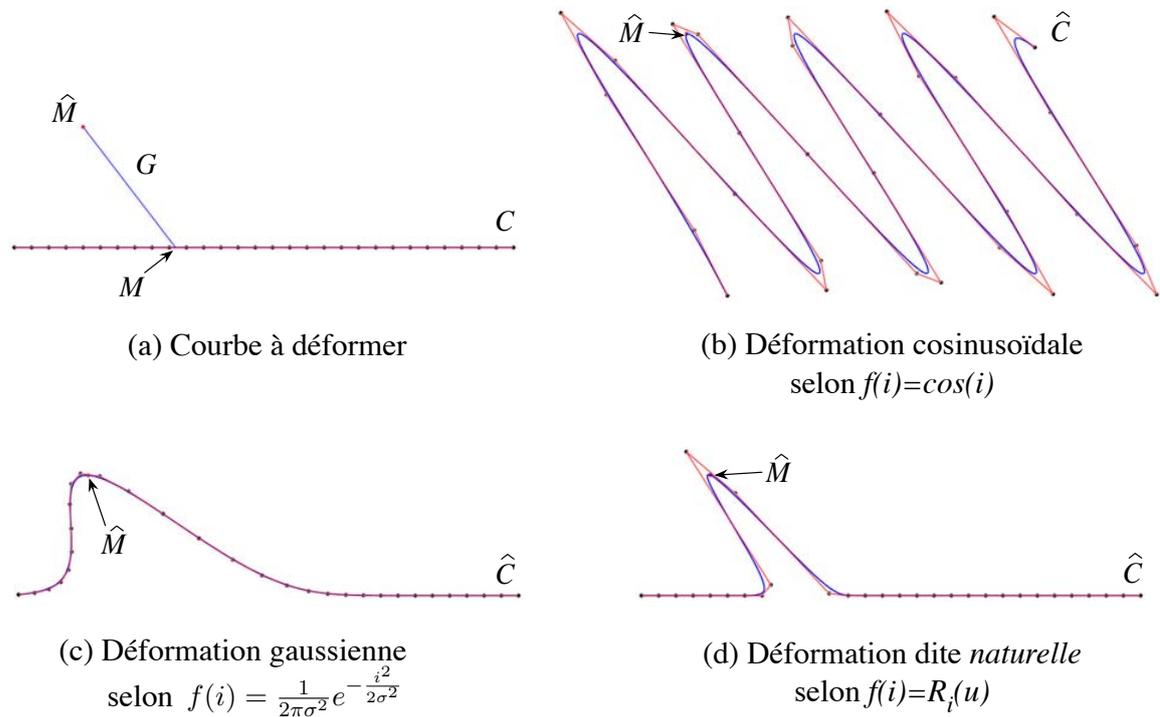


Figure 4.53. Satisfaction d'une même contrainte spatiale selon trois contraintes de localité sur une courbe de degré 3, définie par 30 points de contrôle. L'écart type utilisé dans la déformation gaussienne est $\sigma = 3$

Afin de respecter l'influence «naturelle» et locale des points de contrôle de la courbe \mathcal{C} , un choix *naturel* de fonction de localisation est $f(i) = R_i(u)$ (Fig. 4.53.d). Ainsi, la formulation (4.10) devient :

$$\overrightarrow{m(i)} = \frac{R_i(u)}{\sum_{k=0}^{n-1} [R_k(u)^2]} \vec{e} \quad (4.13)$$

Par ce choix, seulement les $p+1$ points de contrôle définissant \mathcal{C} au paramètre u sont déplacés (Fig. 4.53.d). Il peut cependant être intéressant de contrôler un intervalle de déformation sans être contraint par le degré p de la courbe. Ainsi, afin d'étendre l'influence des points de contrôle de \mathcal{C} , nous proposons de considérer $R_i(u)$ ($i \in [0, n-1]$) comme étant un signal discret $R(u)$ sur lequel est appliqué une convolution discrète utilisant un quelconque filtre discret h . h doit tout de même respecter les conditions de la remarque 4.3.3. Un nouvel ensemble de coefficients d'influence $f(i) = \tilde{B}_i(u)$ est défini tel que :

$$\tilde{B}(u) = h \otimes R(u) \Leftrightarrow \tilde{B}_i(u) = \sum_{k=-\hat{n}}^{\hat{n}} [h(\hat{n} + k) R_{I(i+k)}(u)] \quad (4.14)$$

où – $I(i)$ est une fonction permettant le calcul du produit de convolution aux extrémités de la courbe en utilisant une répétition des points de contrôle par exemple :

$$I(i) = \begin{cases} \hat{n} & \forall i \in [0, \hat{n} - 1] \\ i & \forall i \in [\hat{n}, n - \hat{n} - 1] \\ n - \hat{n} - 1 & \forall i \in [n - \hat{n}, n - 1] \end{cases}$$

– h est un masque de convolution [Cocquerez et Philipp, 1998] de taille $2\hat{n} + 1$ où \hat{n} est son rayon d'influence. Un filtre passe bas comme un filtre gaussien (Fig. 4.54) par exemple peut être utilisé.

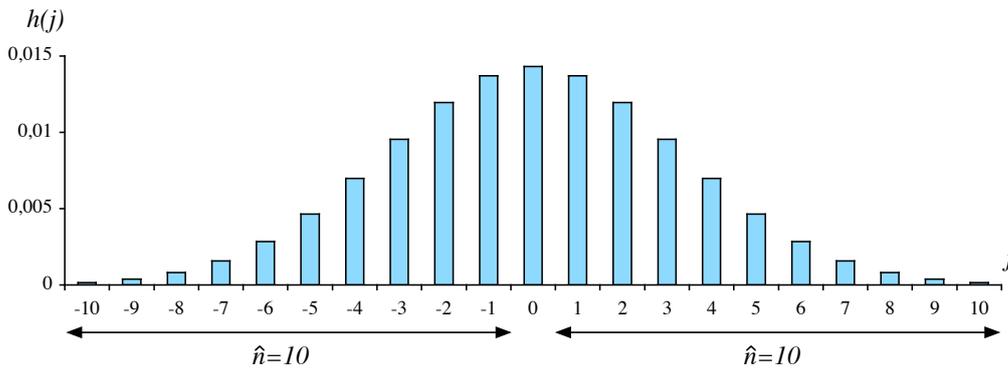


Figure 4.54. Exemple de filtre gaussien avec $\hat{n} = 10$

La figure 4.55 illustre l'impact de la taille d'un filtre gaussien sur une même contrainte en utilisant une courbe de degré 3 définie par 30 points de contrôle. Le schéma 4.55.a présente la courbe originale, sa contrainte et un histogramme représentant l'influence

naturelle, c'est-à-dire les valeurs de $R_i(u)$, des pôles sur le point M de paramètre u . Les schémas 4.55.b, 4.55.c et 4.55.d donnent le résultat de la satisfaction de la contrainte accompagné de l'histogramme représentant les coefficients de $f(i)$ après la convolution de $R_i(u)$ avec un filtre gaussien de rayon $\hat{n} = 5$, $\hat{n} = 15$ puis $\hat{n} = 25$.

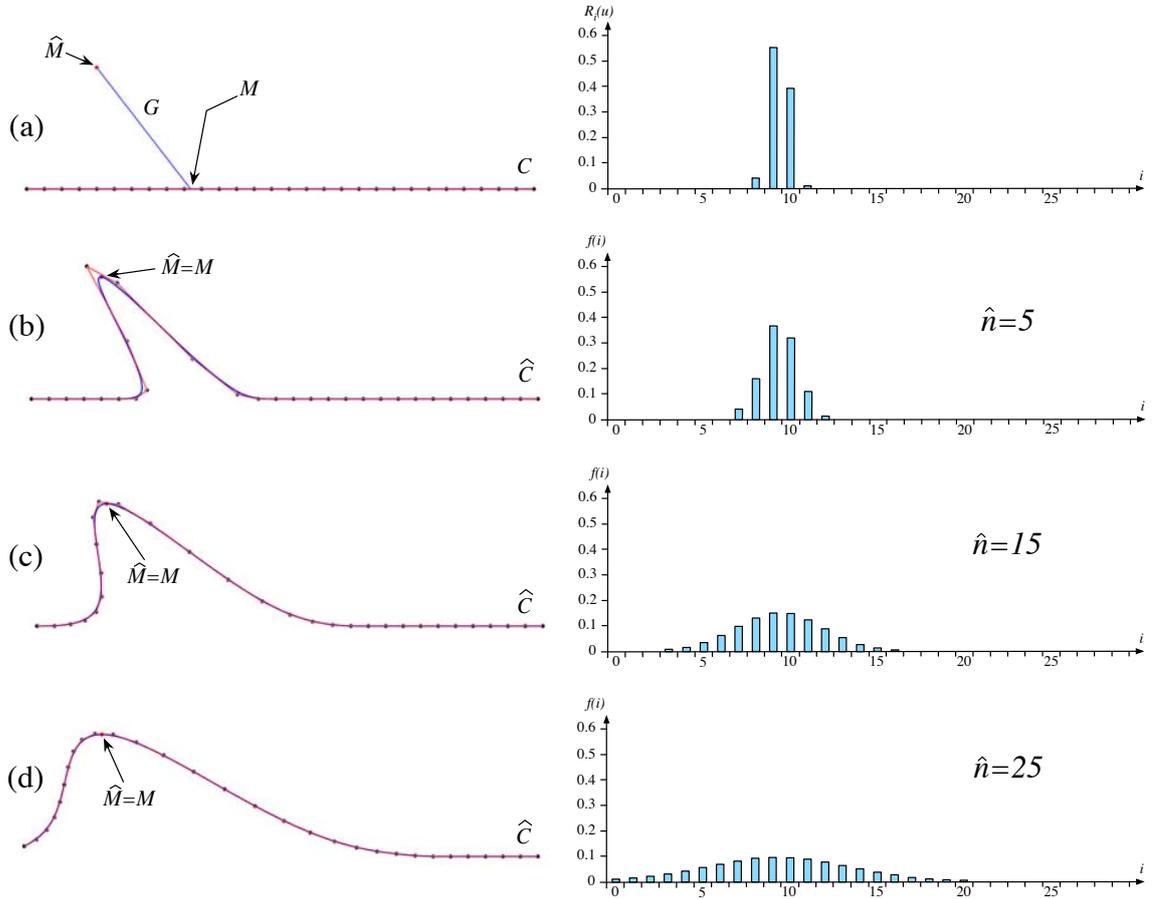


Figure 4.55. Influence de plusieurs filtres gaussiens sur une même contrainte appliquée sur une courbe de degré 3 et définie par 30 points de contrôle

Par cette méthode, l'influence de chaque point de contrôle P_i est étendue en fonction de la taille et de la forme du masque h utilisé :

$$\overrightarrow{m(i)} = \frac{\tilde{B}_i(u)}{\sum_{k=0}^{n-1} [R_k(u) \tilde{B}_k(u)]} \overrightarrow{e} \quad (4.15)$$

Remarque 4.3.4 – *Le processus de convolution peut être réalisé plus rapidement si l'on prend en compte la nature du masque utilisé. [Waltz and Miller, 1997] propose une méthode efficace utilisant un automate à états finis pour appliquer une convolution impliquant un masque gaussien.*

Remarque 4.3.5 – Le produit de convolution $h \otimes R(u)$ peut prendre un grand temps de calcul si le masque utilisé est de grande taille. Afin d'accélérer ce processus, il est intéressant d'utiliser une propriété fondamentale du traitement des signaux : «le procédé de convolution dans le domaine spatial est équivalent à une multiplication dans le domaine des fréquences». La transformée de Fourier rapide¹⁰ [Nussbaumer, 1982] est utilisée pour convertir les données discrètes issues du domaine spatial vers le domaine des fréquences et vice-versa (FFT^{-1}). Ainsi :

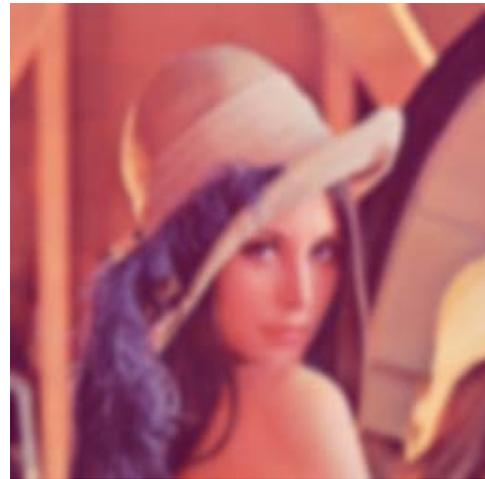
$$\tilde{B}(u) = FFT^{-1}[\tilde{R}(u)]$$

où - $\tilde{R}_i(u) = \hat{h}_i \times FFT[R(u)]_i$;
 - \hat{h} , de taille n , peut être, par exemple, défini par un filtre gaussien de fréquences de rayon \hat{n} .

Ce principe de filtrage est très couramment utilisé en traitement des images pour les rendre plus floues ou au contraire pour en faire ressortir leurs contours. Dans ce domaine aussi, les deux techniques de convolution que nous venons de voir sont utilisées. La figure 4.56 présente un exemple de filtrage flou par convolution matricielle utilisant un masque gaussien de rayon (en pixels) $\hat{n} = 2$ sur l'image de *Léna* classiquement présentée en traitement de l'image.



(a) *Léna* original



(b) *Léna* après filtrage

Figure 4.56. Filtrage de l'image de *Léna* par convolution matricielle utilisant un filtre gaussien de rayon (en pixels) $\hat{n} = 2$

Cette méthode est totalement fonctionnelle pour une contrainte sur les courbes. Le passage aux surfaces est présenté au paragraphe suivant, où l'on peut, cette fois-ci, considérer l'influence de la grille de contrôle comme une image sur laquelle il est possible d'effectuer des filtrages comme nous venons de le voir avec la figure 4.56.

¹⁰aussi connue sous le nom de Fast Fourier Transform ou *FFT*

4.3.5.2 Application aux surfaces NURBS

La méthode expliquée précédemment peut très facilement être appliquée aux surfaces (Fig. 4.57). Il suffit pour cela de voir l'espace paramétrique comme un plan, la contrainte paramétrique s'effectue donc sur un couple (u, v) . Le polyèdre de contrôle doit être considéré comme une grille et la contrainte de localité comme une fonction $f(i, j)$ de l'espace à deux dimensions telle que :

$$\begin{cases} f : [0, n-1] \times [0, m-1] \mapsto \mathbb{R}^+ \\ \exists (i, j) \in [0, n-1] \times [0, m-1] / R_{i,j}(u, v)f(i, j) \neq 0 \end{cases}$$

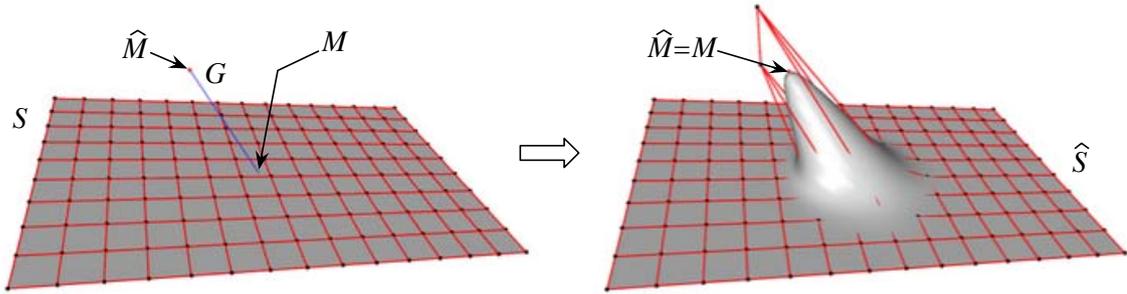


Figure 4.57. Contrainte ponctuelle de passage sur une surface de degrés 3×3 et définie par 15×10 points de contrôle (déformation *naturelle*)

En s'appuyant sur les définitions et les notations de la partie 1.3.4.2, la proposition 4.3.1 devient :

Proposition 4.3.2 : Soit $\vec{e} = \overrightarrow{MM'}$, le vecteur déplacement $\overrightarrow{m(i, j)}$ à appliquer au point de contrôle $P_{i,j}$ de S pour satisfaire la contrainte \mathcal{G} au point de paramètres (u, v) s'écrit de la manière suivante :

$$\overrightarrow{m(i, j)} = \frac{f(i, j)}{\sum_{k=0}^{n-1} \sum_{l=0}^{m-1} [R_{k,l}(u, v)f(k, l)]} \vec{e} \quad (4.16)$$

Preuve :

Soit O l'origine du repère.

$$\begin{aligned} \overrightarrow{OS(u, v)} &= \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [R_{i,j}(u, v) \overrightarrow{OP_{i,j}}] && \text{avec } \overrightarrow{OP_{i,j}} = \overrightarrow{OP_{i,j}} + \overrightarrow{m(i, j)} \\ &= \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [R_{i,j}(u, v) (\overrightarrow{OP_{i,j}} + \overrightarrow{m(i, j)})] \\ &= \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [R_{i,j}(u, v) \overrightarrow{OP_{i,j}}] + \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [R_{i,j}(u, v) \overrightarrow{m(i, j)}] \\ &= \overrightarrow{OS(u, v)} + \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [R_{i,j}(u, v) \overrightarrow{m(i, j)}] \end{aligned} \quad (4.17)$$

En utilisant (4.16) on obtient immédiatement :

$$\overrightarrow{O\hat{S}(u,v)} = \overrightarrow{OS(u,v)} + \vec{e} = \overrightarrow{OM} \quad (4.18)$$

□

Les paramètres (u, v) de la contrainte G sont définis de la même manière que sur les courbes (Cf. 4.3.5.1.1).

Comme cela a été précisé dans la remarque 4.3.3, $f(i, j)$ est définie sur \mathbb{R}^+ afin de garantir que le dénominateur de (4.16) soit strictement positif (avec un poids positif ou nul attaché à chaque point de contrôle). Mais toute fonction $f(i, j)$ est valide sous réserve de vérifier :

$$\sum_{k=0}^{n-1} \sum_{l=0}^{m-1} [R_{k,l}(u, v) f(k, l)] \neq 0$$

Il est donc possible de satisfaire une contrainte d'influence $f(i, j) = \cos(i)$ (Fig. 4.58.a) de la même manière qu'une contrainte ayant pour influence un potentiel gaussien. Ce dernier cas qui se présente sous la forme de $f(i, j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$, est illustré sur la figure 4.58.b en utilisant $\sigma = 2$.

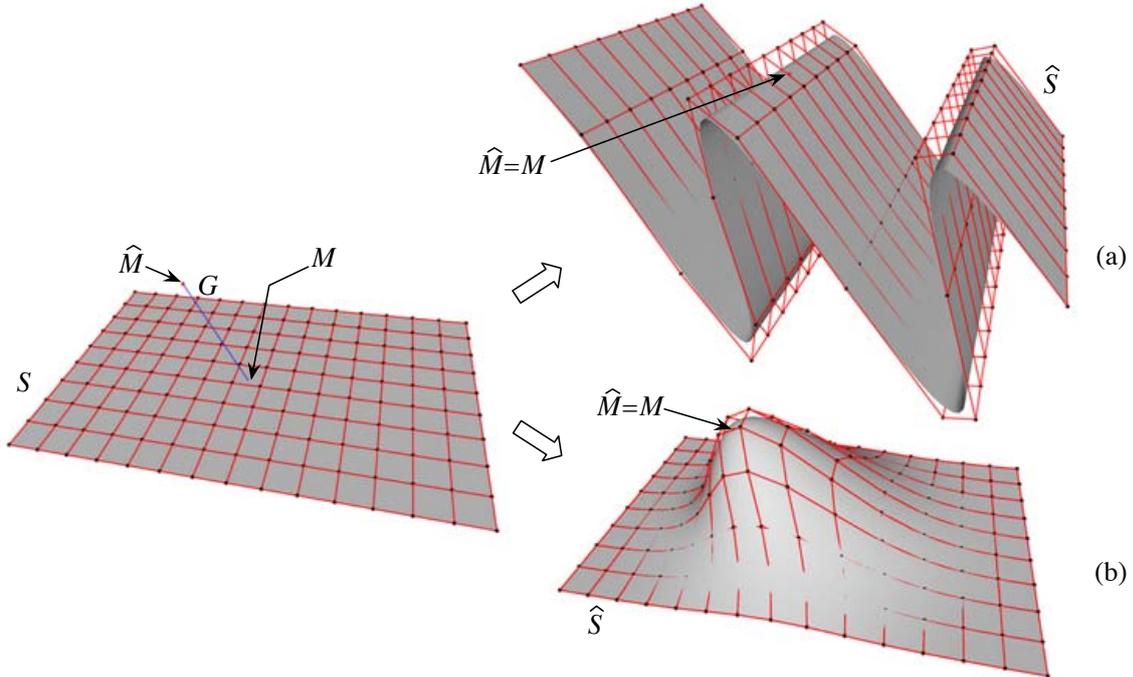


Figure 4.58. Deux types d'influence pour une même contrainte ponctuelle de passage sur une surface de degrés 3×3 définie par 15×10 points de contrôle

Dans le cas d'une influence dite *naturelle* (Fig. 4.57), la contrainte de localité se définit par $f(i, j) = R_{i,j}(u, v)$ et la formulation (4.16) devient :

$$\overrightarrow{m(i,j)} = \frac{R_{i,j}(u, v)}{\sum_{k=0}^{n-1} \sum_{l=0}^{m-1} [R_{k,l}(u, v)^2]} \vec{e} \quad (4.19)$$

En considérant $R_{i,j}(u, v)$, avec $(i, j) \in [0, n-1] \times [0, m-1]$, comme un signal discret $R(u, v)$ au même titre que le serait une image de taille $n \times m$, il est possible d'introduire un nouvel ensemble de coefficients d'influence $f(i, j) = \tilde{B}_{i,j}(u, v)$ défini par :

$$\tilde{B}(u, v) = h \otimes R(u, v) \Leftrightarrow \tilde{B}_{i,j}(u, v) = \sum_{k=-\hat{n}}^{\hat{n}} \sum_{l=-\hat{n}}^{\hat{n}} [h(\hat{n} + k, \hat{n} + l) R_{I(i+k), J(j+l)}(u, v)] \quad (4.20)$$

où – $I(i)$ et $J(j)$ sont des fonctions permettant le calcul du produit de convolution aux extrémités de la surface en utilisant une répétition des points de contrôle par exemple :

$$I(i) = \begin{cases} \hat{n} & \forall i \in [0, \hat{n} - 1] \\ i & \forall i \in [\hat{n}, n - \hat{n} - 1] \\ n - \hat{n} - 1 & \forall i \in [n - \hat{n}, n - 1] \end{cases} \quad J(j) = \begin{cases} \hat{n} & \forall j \in [0, \hat{n} - 1] \\ j & \forall j \in [\hat{n}, m - \hat{n} - 1] \\ m - \hat{n} - 1 & \forall j \in [m - \hat{n}, m - 1] \end{cases}$$

- h est un masque de convolution de taille $2\hat{n} + 1 \times 2\hat{n} + 1$ où \hat{n} est son rayon d'influence. Comme pour les courbes, un filtre gaussien peut être utilisé pour propager l'influence naturelle des pôles. La matrice de convolution d'un tel filtre peut être la suivante pour $\hat{n} = 3$:

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 6 & 10 & 6 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \Rightarrow \text{filtre gaussien en représentation entière (somme=62)}$$

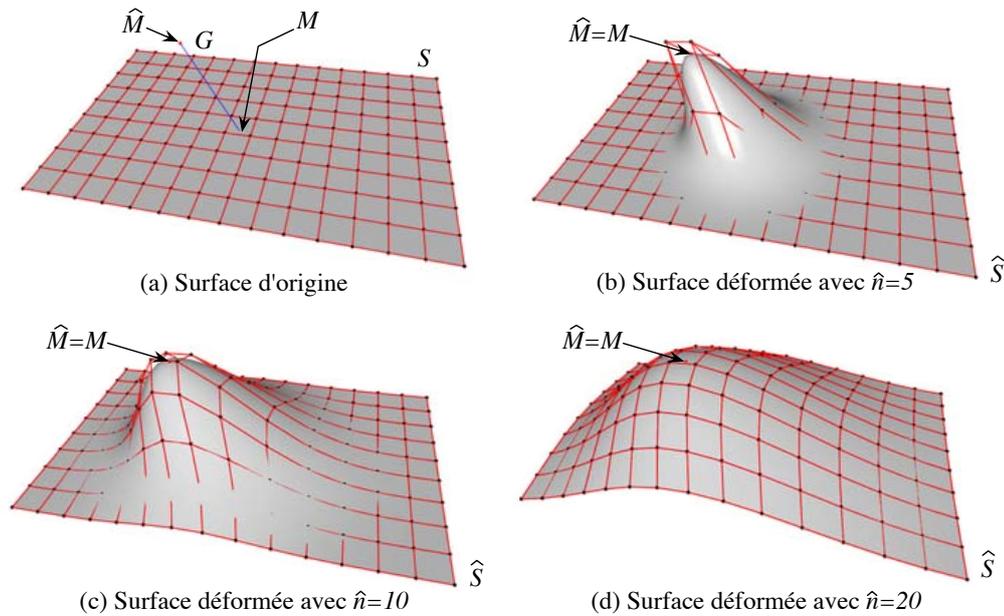


Figure 4.59. Satisfaction d'une même contrainte sur une surface en utilisant plusieurs masques gaussiens pour filtrer l'influence *naturelle* des pôles

La figure 4.59 présente l'influence de trois masques de type gaussien de rayons différents pour la satisfaction d'une même contrainte ponctuelle de passage. La surface d'origine initialement plane (degrés 3×3 pour 15×10 pôles), est illustrée sur la figure 4.59.a. La figure 4.59.b est le résultat de la déformation réalisée avec le rayon du masque $\hat{n} = 5$, la figure 4.59.c correspond à $\hat{n} = 10$ et la figure 4.59.d à $\hat{n} = 20$.

Par cette technique, il est possible de déformer une surface pour satisfaire une contrainte \mathcal{G} donnée tout en contrôlant son influence à partir d'une fonction $f(i, j)$. Cependant, cette méthode ne fonctionne que pour une contrainte ponctuelle de passage. C'est pourquoi, dans la partie suivante nous abordons le cas de plusieurs contraintes de passages à résoudre portant sur une même courbe que nous étendons ensuite aux surfaces.

4.3.6 Généralisation à N contraintes ponctuelles avec contrôle d'influence

En pratique, plus d'une contrainte est à satisfaire. Nous proposons pour résoudre ce problème, deux algorithmes itératifs puis une approche algébrique. Ces méthodes sont en premier lieu développées pour les courbes paramétriques puis nous donnons les formulations utilisées pour l'extension aux surfaces.

4.3.6.1 Application aux courbes

4.3.6.1.1 Notations

- Nous considérons N contraintes G_λ ($\lambda \in [0, N - 1]$) chacune définie à un paramètre $u_\lambda \in [U_0, U_{p+n}]$ avec une fonction d'influence $f_\lambda(i)$. Chaque G_λ doit déformer \mathcal{C} pour obtenir $\mathcal{C}(u_\lambda) = \hat{M}_\lambda$. Il est à remarquer que $N \leq n$ est une condition nécessaire pour obtenir une solution ;
- Soit $\overrightarrow{e_{\lambda,t}} = \overrightarrow{M_{\lambda,t} \hat{M}_\lambda}$ le vecteur d'erreur associé à G_λ après t itérations ($t > 0$) et $\mathcal{C}_t(u_\lambda) = M_{\lambda,t}$. Le vecteur erreur initial est $\overrightarrow{e_\lambda} = \overrightarrow{e_{\lambda,0}} = \overrightarrow{M_{\lambda,0} \hat{M}_\lambda} = \overrightarrow{M_\lambda \hat{M}_\lambda}$;
- Soit $E_t = \sum_{\lambda=0}^{N-1} \|\overrightarrow{e_{\lambda,t}}\|_2$ l'erreur totale après t itérations. Nous avons considéré la norme euclidienne ($\|\cdot\|_2$) définie par $\|(x_1, x_2, \dots, x_n)\|_k = \left(\sum_{l=1}^n |x_l|^k \right)^{1/k}$ avec $k = 2$ comme bien adaptée pour estimer l'erreur globale à la fin de chaque itération ;
- Considérons N vecteurs $\overrightarrow{R(\lambda)}$ de taille n :

$$\overrightarrow{R(\lambda)} = \begin{pmatrix} R_0(u_\lambda) \\ R_1(u_\lambda) \\ \vdots \\ R_{n-1}(u_\lambda) \end{pmatrix}$$

$R(\lambda)$ est ainsi défini à partir des fonctions B-Spline attachées à la contrainte G_λ ;

Proposition 4.3.3 : $\forall \lambda \in [0, N - 1]$ les vecteurs $\overrightarrow{R(\lambda)}$ sont linéairement indépendants si et seulement si :

$$U_\lambda \leq u_\lambda \leq U_{\lambda+p+1} \quad (0 \leq \lambda \leq n - 1).$$

Preuve :

Considérons le cas limite où $N = n$ et définissons Q comme une matrice $N \times N$ telle que $Q^T = (\overrightarrow{R(0)}, \overrightarrow{R(1)}, \dots, \overrightarrow{R(N-1)})$. Prouver que les vecteurs $\overrightarrow{R(\lambda)}$ sont linéairement indépendants est équivalent à prouver que Q est une matrice inversible sachant que Q est une matrice inversible $\Leftrightarrow Q^T$ est une matrice inversible.

Ainsi, soit $\lambda_0, \dots, \lambda_{n-1}$ définis tels que $\lambda_0 R_0(u_\lambda) + \lambda_1 R_1(u_\lambda) + \dots + \lambda_{n-1} R_{n-1}(u_\lambda) = 0$. Soit $r(u) = \lambda_0 R_0(u) + \dots + \lambda_{n-1} R_{n-1}(u)$, il faut résoudre $\forall \lambda \in [0, N - 1] : r(u_\lambda) = 0$.

$$\begin{aligned} r(u_\lambda) &= \lambda_0 \frac{w_0 N_0(u_\lambda)}{\sum_{k=0}^{n-1} w_k N_k(u_\lambda)} + \dots + \lambda_{n-1} \frac{w_{n-1} N_{n-1}(u_\lambda)}{\sum_{k=0}^{n-1} w_k N_k(u_\lambda)} \\ &= \frac{1}{\sum_{k=0}^{n-1} w_k N_k(u_\lambda)} (\lambda_0 w_0 N_0(u_\lambda) + \dots + \lambda_{n-1} w_{n-1} N_{n-1}(u_\lambda)) = 0 \end{aligned}$$

Par définition, $\forall \lambda \in [0, N - 1]$ et $\forall i \in [0, n - 1] : N_i(u_\lambda) \neq 0$. En utilisant le théorème de Schoenberg-Whitney [De Boor, 1978], nous savons que $\overrightarrow{N(\lambda)}$ vecteurs définis par $\overrightarrow{N(\lambda)}^T = (R_0(u_\lambda), R_1(u_\lambda), \dots, R_{n-1}(u_\lambda))$ sont linéairement indépendants si et seulement si $U_\lambda \leq u_\lambda \leq U_{\lambda+p+1}$ ($0 \leq j \leq n - 1$).

Donc $\lambda_0 w_0 N_0(u_\lambda) + \dots + \lambda_{n-1} w_{n-1} N_{n-1}(u_\lambda) = 0$ si et seulement si $\forall i \in [0, n - 1]$, $\lambda_i w_i = 0$ et $w_i \neq 0 \Rightarrow \lambda_i = 0$.

□

- Soit $P_{i,t}$ le point de contrôle i au début de l'itération t ;
- Soit $\hat{P}_{i,t} = P_{i,t+1}$ le point de contrôle i après l'itération t ;
- Définissons N vecteurs $\overrightarrow{\tilde{B}(\lambda)}$ de taille n . Où chaque $\tilde{B}_i(\lambda) = f_\lambda(i)$ est l'influence¹¹ de G_λ sur le point de contrôle P_i tel que :

$$\overrightarrow{\tilde{B}(\lambda)} = \begin{pmatrix} \tilde{B}_0(u_\lambda) \\ \tilde{B}_1(u_\lambda) \\ \vdots \\ \tilde{B}_{n-1}(u_\lambda) \end{pmatrix} = \begin{pmatrix} f_\lambda(0) \\ f_\lambda(1) \\ \vdots \\ f_\lambda(n-1) \end{pmatrix}$$

- Soit alors $\overrightarrow{B(\lambda)}$ tel que : $B_i(\lambda) = \frac{|\tilde{B}_i(\lambda)|}{\|\overrightarrow{\tilde{B}(\lambda)}\|_1}$.

Cette formulation de $\overrightarrow{B(\lambda)}$ lui permet d'acquérir les propriétés suivantes :

- 1) $B_i(\lambda) \geq 0 \quad \forall i \in [0, n - 1]$ (non négativité) ;

¹¹ $\tilde{B}_i(\lambda) = R_i(\lambda)$ correspond donc à la déformation naturelle

$$2) \sum_{i=0}^{n-1} B_i(\lambda) = 1 ; \|\overrightarrow{B(\lambda)}\|_1 = 1 \Rightarrow \|\overrightarrow{B(\lambda)}\|_2 \leq 1.$$

Remarque 4.3.6 – Le cas particulier où $\overrightarrow{B(\lambda)} = \overrightarrow{R(\lambda)}$ correspond à la déformation naturelle.

– Soit D la matrice $N \times N$ définie par :

$$D_{k,l} = \langle \overrightarrow{R(k)}, \overrightarrow{B(l)} \rangle = \sum_{i=0}^{n-1} [R_i(k)B_i(l)] = \|\overrightarrow{R(k)}\|_2 \|\overrightarrow{B(l)}\|_2 \cos(\overrightarrow{R(k)}, \overrightarrow{B(l)})$$

$$D = \begin{pmatrix} \langle \overrightarrow{R(0)}, \overrightarrow{B(0)} \rangle & \langle \overrightarrow{R(0)}, \overrightarrow{B(1)} \rangle & \cdots & \langle \overrightarrow{R(0)}, \overrightarrow{B(N-1)} \rangle \\ \langle \overrightarrow{R(1)}, \overrightarrow{B(0)} \rangle & \langle \overrightarrow{R(1)}, \overrightarrow{B(1)} \rangle & \cdots & \langle \overrightarrow{R(1)}, \overrightarrow{B(N-1)} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \overrightarrow{R(N-1)}, \overrightarrow{B(0)} \rangle & \langle \overrightarrow{R(N-1)}, \overrightarrow{B(1)} \rangle & \cdots & \langle \overrightarrow{R(N-1)}, \overrightarrow{B(N-1)} \rangle \end{pmatrix}$$

Remarque 4.3.7 – $0 < D_{k,l} \leq 1$ puisque $\overrightarrow{R(k)}$ et $\overrightarrow{B(l)}$ ont des coefficients positifs ou nuls, inférieurs ou égaux à 1. On peut aussi dire que $\|\overrightarrow{R(k)}\|_2 \leq 1$ et $\|\overrightarrow{B(l)}\|_2 \leq 1$. En effet, on sait, par définition que $\sum_{i=0}^{n-1} R_i(k) = 1$, donc $\left(\sum_{i=0}^{n-1} R_i(k)\right)^2 = 1$, or $\forall i \in [0, n-1] R_i(k) \geq 0$, donc $\sum_{i=0}^{n-1} R_i^2(k) \leq \left(\sum_{i=0}^{n-1} R_i(k)\right)^2 = 1$, ainsi on a bien $\sqrt{\sum_{i=0}^{n-1} R_i^2(k)} \leq 1$. La méthode est la même pour $\overrightarrow{B(l)}$ puisque $\sum_{i=0}^{n-1} B_i(l) = 1$ avec $\forall i \in [0, n-1] B_i(l) \geq 0$.

– Soit $\overrightarrow{m_t(i)}$ le déplacement appliqué à $P_{i,t}$ tel que :

$$\overrightarrow{OP_{i,t}} = \overrightarrow{OP_{i,t}} + \overrightarrow{m_t(i)} = \overrightarrow{OP_{i,t+1}} \Leftrightarrow \hat{P}_{i,t} = P_{i,t+1}$$

4.3.6.1.2 Méthode 1 : Approche globale

La première méthode que nous proposons est dite *totale* car elle consiste à déplacer tous les points de contrôle, dans l'influence des contraintes, à chaque itération. Le déplacement appliqué à chaque itération a pour objectif de faire décroître la distance entre la courbe et toutes les contraintes. Ce processus itératif est stoppé lorsque la précision souhaitée est atteinte. Le déplacement appliqué à chaque point de contrôle P_i à chaque itération t est :

$$\overrightarrow{m_t(i)} = \frac{1}{N} \sum_{\lambda=0}^{N-1} B_i(\lambda) \overrightarrow{e_{\lambda,t-1}} \quad \text{avec } t > 0 \quad (4.21)$$

Le processus itératif est arrêté, lorsque E_t est considérée assez petite en fonction des besoins de l'utilisateur. Il peut aussi être arrêté, lorsque la différence d'erreur δE entre deux itérations est considérée comme petite. Cette notion est à discrétion de l'utilisateur. L'algorithme 2, consultable en annexe D.2, implémente cette méthode et la figure 4.60 illustre son déroulement sur une courbe B-Spline de degré 3 définie par 5 points de contrôle.

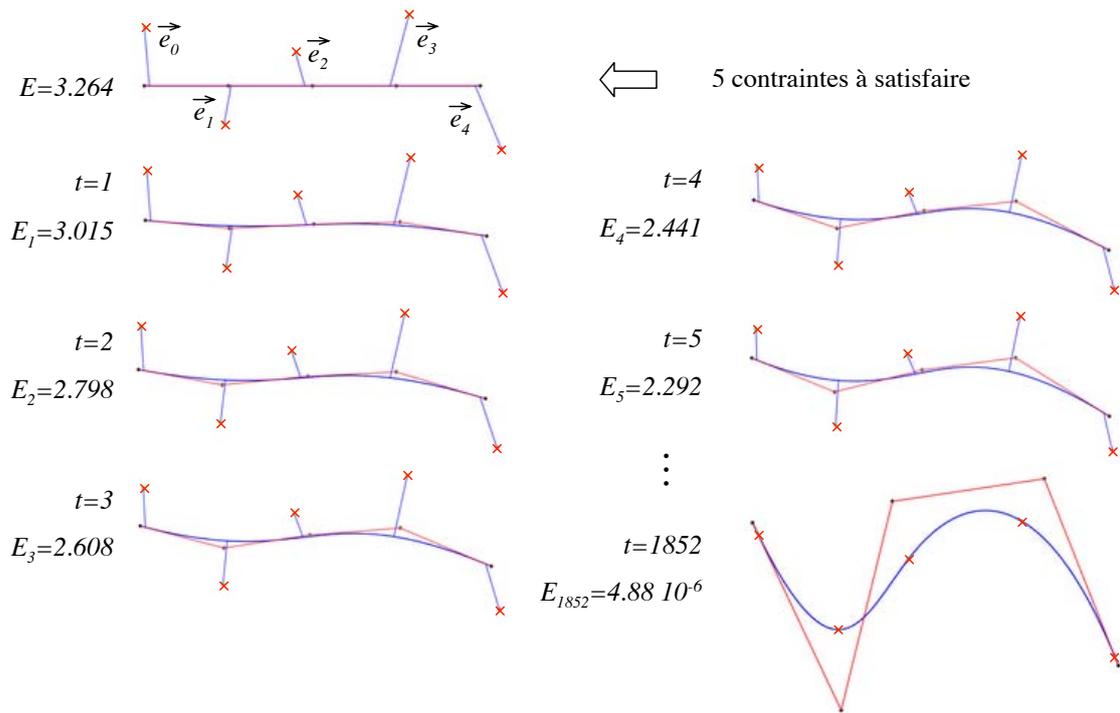


Figure 4.60. Déroulement pas à pas de la méthode globale sur une courbe B-Spline de degré 3 définie par 5 pôles (140 ms sur un P4 - 2.4 Ghz - 1.00 Go RAM)

Bien que les résultats pratiques soient cohérents et que le comportement de la méthode soit stable, sa convergence n'est, à l'heure actuelle, pas démontrée pour le cas général. Ce travail fait partie des perspectives rattachées à cette méthode. En pratique, et bien que ce ne soit pas démontré dans ce document, la méthode converge pour toute contrainte ayant une influence $f(i)$ respectant la remarque 4.3.3. La vitesse de convergence de la méthode dépend de la proximité paramétrique des contraintes et du recouplement de leur influence. Au plus les influences sont en interaction (Fig. 4.61), au plus la convergence est lente. Si une interaction est totale, on est dans le cas d'un problème surcontraint.

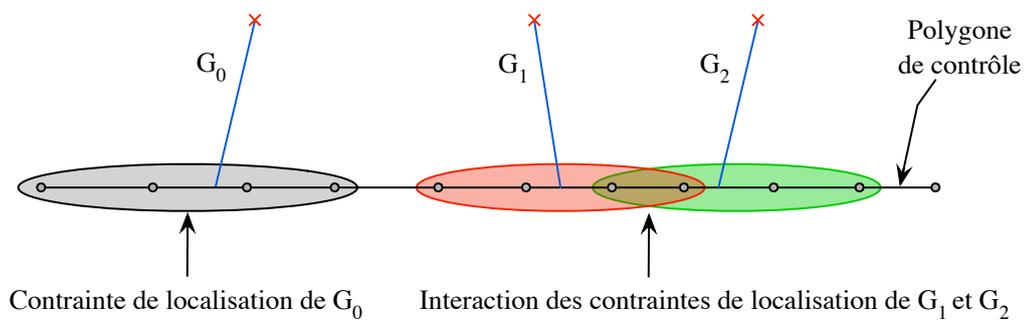


Figure 4.61. Interaction des contraintes de localisation de type influence *naturelle*, sur une courbe B-Spline de degré 3

Remarque 4.3.8 – Lors de nos tests, il s'avère que dans le cas d'un problème surcontraint, la courbe se déforme jusqu'à converger vers une courbe passant au plus près de l'ensemble des contraintes spatiales. Ainsi même si plusieurs contraintes ont la même contrainte paramétrique avec des contraintes spatiales différentes, la courbe converge vers le barycentre des contraintes spatiales comme cela est illustré sur la figure 4.62. Nous avons pour perspective de démontrer ce comportement.

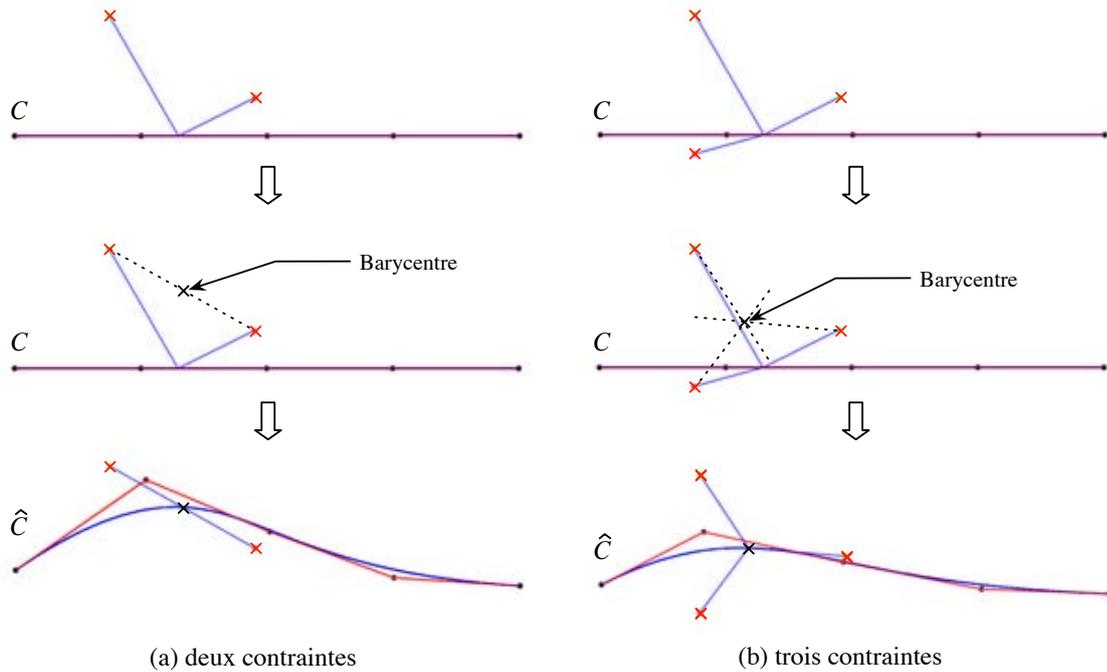


Figure 4.62. Cas, appliqué sur une courbe, où des contraintes d'influence *naturelle* ont la même contrainte paramétrique pour des contraintes spatiales différentes

Nous avons prouvé la convergence de cette méthode dans le cas non surcontraint où l'influence des contraintes est dite *naturelle* ($\overrightarrow{B(\lambda)} = \overrightarrow{R(\lambda)}$). En effet, la méthode peut aussi s'exprimer sous une forme algébrique. Résoudre ce système algébrique revient à parfaitement satisfaire l'ensemble des contraintes. Nous avons démontré que ce système peut être résolu dans le cas de l'influence *naturelle* des contraintes, c'est ce qui est expliqué dans le paragraphe suivant.

4.3.6.1.3 Expression algébrique de la méthode globale

L'expression réursive (4.21) peut être reformulée en une expression algébrique.

Proposition 4.3.4 : Le vecteur erreur attaché à la contrainte G_λ après t itérations peut être calculé selon le vecteur erreur après $t - 1$ itérations comme suit :

$$\vec{e}_{\lambda,t} = \vec{e}_{\lambda,t-1} - \sum_{k=0}^{N-1} \frac{1}{N} \langle \vec{R}(\lambda), \vec{B}(k) \rangle \vec{e}_{k,t-1} \quad (4.22)$$

Preuve :

Soit O l'origine du repère.

$$\begin{aligned} \vec{e}_{\lambda,t} &= \vec{M}_{\lambda,t} \hat{M}_\lambda = \vec{C}_t(u_\lambda) \hat{M}_\lambda = \vec{O} \hat{M}_\lambda - \sum_{i=0}^{n-1} R_i(\lambda) \vec{O} P_{i,t} \\ &= \vec{O} \hat{M}_\lambda - \sum_{i=0}^{n-1} R_i(\lambda) \left[\vec{O} P_{i,t-1} + \vec{m}_{t-1}(i) \right] \\ &= \vec{O} \hat{M}_\lambda - \sum_{i=0}^{n-1} R_i(\lambda) \vec{O} P_{i,t-1} - \sum_{i=0}^{n-1} R_i(\lambda) \frac{1}{N} \sum_{k=0}^{N-1} B_i(k) \vec{e}_{k,t-1} \\ \text{et } \vec{O} \hat{M}_\lambda - \sum_{i=0}^{n-1} R_i(\lambda) \vec{O} P_{i,t-1} &= \vec{C}_{t-1}(u_\lambda) \hat{M}_\lambda = \vec{e}_{\lambda,t-1} \end{aligned}$$

Ainsi $\vec{e}_{\lambda,t} = \vec{e}_{\lambda,t-1} - \sum_{k=0}^{N-1} \frac{1}{N} \sum_{i=0}^{n-1} R_i(\lambda) B_i(k) \vec{e}_{k,t-1}$ ce qui correspond exactement à (4.22)

□

Soit S une matrice $N \times N$ définie par $S = I - \frac{1}{N} D$:

$$S = \begin{pmatrix} 1 - \frac{1}{N} \langle \vec{R}(0), \vec{B}(0) \rangle & -\frac{1}{N} \langle \vec{R}(0), \vec{B}(1) \rangle & \cdots & -\frac{1}{N} \langle \vec{R}(0), \vec{B}(N-1) \rangle \\ -\frac{1}{N} \langle \vec{R}(1), \vec{B}(0) \rangle & 1 - \frac{1}{N} \langle \vec{R}(1), \vec{B}(1) \rangle & \cdots & -\frac{1}{N} \langle \vec{R}(1), \vec{B}(N-1) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{N} \langle \vec{R}(N-1), \vec{B}(0) \rangle & -\frac{1}{N} \langle \vec{R}(N-1), \vec{B}(1) \rangle & \cdots & 1 - \frac{1}{N} \langle \vec{R}(N-1), \vec{B}(N-1) \rangle \end{pmatrix}$$

Les valeurs de S sont telles que : $\begin{cases} 0 < S_{\lambda,\lambda} < 1 & \text{si } \lambda = k \\ -\frac{1}{N} < S_{\lambda,k} < 0 & \text{ailleurs} \end{cases}$

L'équation (4.22) devient donc :

$$\vec{e}_{\lambda,t} = \sum_{k=0}^{N-1} S_{\lambda,k} \vec{e}_{k,t-1} \quad (4.23)$$

Proposition 4.3.5 : Le vecteur erreur de l'itération t peut être directement calculé de la façon suivante :

$$\vec{e}_{\lambda,t} = \sum_{k=0}^{N-1} [S^t]_{\lambda,k} \vec{e}_k \quad (4.24)$$

Preuve :

Soit $t > 0$. En utilisant (4.23), la preuve est faite par récurrence sur t .

- Cas où $t = 1$: $\overrightarrow{e_{\lambda,1}} = \sum_{k=0}^{N-1} S_{\lambda,k} \overrightarrow{e_{k,0}} = \sum_{k=0}^{N-1} S_{\lambda,k} \overrightarrow{e_k}$
- Si l'hypothèse (4.24) est vérifiée pour t , l'est-elle encore pour $t + 1$?

$$\overrightarrow{e_{\lambda,t+1}} = \sum_{k=0}^{N-1} S_{\lambda,k} \overrightarrow{e_{k,t}}$$
 en utilisant l'hypothèse (4.24)

$$= \sum_{l=0}^{N-1} S_{\lambda,l} \sum_{k=0}^{N-1} [S^t]_{l,k} \overrightarrow{e_k} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} S_{\lambda,l} [S^t]_{l,k} \overrightarrow{e_k} = \sum_{k=0}^{N-1} [S^{t+1}]_{\lambda,k} \overrightarrow{e_k}$$

□

En utilisant (4.21), le déplacement appliqué au point de contrôle P_i à l'itération $t > 0$ est :

$$\overrightarrow{m_t(i)} = \frac{1}{N} \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} [S^{t-1}]_{\lambda,k} \overrightarrow{e_k} \quad (4.25)$$

Ainsi le déplacement total appliqué à P_i est :

$$\begin{aligned} \overrightarrow{m(i)} &= \lim_{t \rightarrow +\infty} \sum_{l=1}^t \overrightarrow{m_l(i)} = \lim_{t \rightarrow +\infty} \sum_{l=1}^t \frac{1}{N} \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} [S^{l-1}]_{\lambda,k} \overrightarrow{e_k} \\ &= \frac{1}{N} \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} \left[\lim_{t \rightarrow +\infty} \sum_{l=1}^t S^{l-1} \right]_{\lambda,k} \overrightarrow{e_k} \end{aligned} \quad (4.26)$$

Proposition 4.3.6 : La méthode converge vers la solution correcte (satisfaction de toutes les contraintes) si et seulement si : $\lim_{t \rightarrow +\infty} \sum_{l=1}^t S^{l-1}$ existe et est finie.

Bien que les résultats pratiques tendent à approuver cette proposition, sa preuve n'est, à ce jour, pas disponible dans le cas général. Nous avons cependant prouvé dans la preuve E.1 présentée en annexe, que la proposition 4.3.6 est vraie dans le cas de l'influence *naturelle* des contraintes. Néanmoins, lorsque la proposition 4.3.6 est vraie, la limite de la série géométrique peut être calculée telle que : $\lim_{t \rightarrow \infty} \sum_{l=1}^t [S^{l-1}] = [I - S]^{-1}$

Or $S = I - \frac{1}{N}D \Leftrightarrow \frac{1}{N}D = I - S$ donc :

$$\lim_{t \rightarrow \infty} \sum_{l=1}^t [S^{l-1}] = \left(\frac{1}{N}D \right)^{-1} = ND^{-1} \quad (4.27)$$

En utilisant (4.26) et (4.27) le déplacement local à appliquer au point de contrôle P_i afin de satisfaire l'ensemble des contraintes est :

$$\overrightarrow{m(i)} = \frac{1}{N} \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} [ND^{-1}]_{\lambda,k} \overrightarrow{e_k}$$

Proposition 4.3.7 : Le déplacement à appliquer à chaque point de contrôle P_i de \mathcal{C} afin d'obtenir $\hat{\mathcal{C}}$ vérifiant $\hat{\mathcal{C}}(u_\lambda) = \hat{M}_\lambda \quad \forall \lambda \in [0, N-1]$ est $\overrightarrow{m(i)}$ défini tel que :

$$\overrightarrow{m(i)} = \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} [D^{-1}]_{\lambda,k} \overrightarrow{e_k} \quad (4.28)$$

Preuve :

Soit O l'origine du repère. En partant de l'équation (4.11) et en y injectant (4.28), on a :

$$\begin{aligned} \overrightarrow{O\hat{\mathcal{C}}(u_\lambda)} &= \overrightarrow{OC(u_\lambda)} + \sum_{i=0}^{n-1} [R_i(\lambda) \overrightarrow{m(i)}] \\ &= \overrightarrow{OC(u_\lambda)} + \sum_{i=0}^{n-1} \left[R_i(\lambda) \left(\sum_{l=0}^{N-1} B_i(l) \sum_{k=0}^{N-1} [D^{-1}]_{l,k} \overrightarrow{e_k} \right) \right] \\ &= \overrightarrow{OC(u_\lambda)} + \sum_{l=0}^{N-1} \sum_{i=0}^{n-1} R_i(\lambda) B_i(l) \sum_{k=0}^{N-1} [D^{-1}]_{l,k} \overrightarrow{e_k} \\ &= \overrightarrow{OC(u_\lambda)} + \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} D_{\lambda,l} [D^{-1}]_{l,k} \overrightarrow{e_k} \\ &= \overrightarrow{OC(u_\lambda)} + \sum_{k=0}^{N-1} [DD^{-1}]_{\lambda,k} \overrightarrow{e_k} = \overrightarrow{OC(u_\lambda)} + \sum_{k=0}^{N-1} I_{\lambda,k} \overrightarrow{e_k} \\ &= \overrightarrow{OC(u_\lambda)} + \overrightarrow{e_\lambda} = \overrightarrow{OM_\lambda} \end{aligned}$$

Ce qui correspond exactement à (4.12).

□

Ainsi, l'équation (4.28) permet de calculer directement la position finale de chaque point de contrôle P_i pour satisfaire les N contraintes ponctuelles de passage lorsque cela est possible, c'est-à-dire lorsque D est inversible. Outre cette première méthode qui est globale, nous avons aussi pensé à une autre méthode qui exploite directement les résultats de la partie 4.3.5.1. Elle est présentée dans le paragraphe suivant et une comparaison des deux méthodes ainsi que leurs intérêts sont proposés dans le paragraphe 4.3.6.1.6.

4.3.6.1.4 Méthode 2 : Approche séquentielle

Avec la notation introduite dans la partie 4.3.6.1.1, la formule générale (4.10) utilisée pour satisfaire une contrainte λ nommée G_λ devient :

$$\overrightarrow{m_\lambda(i)} = \frac{\tilde{B}_i(\lambda)}{\sum_{k=0}^{n-1} [R_k(\lambda) \tilde{B}_k(\lambda)]} \overrightarrow{e_\lambda} = \frac{B_i(\lambda)}{\sum_{k=0}^{n-1} [R_k(\lambda) B_k(\lambda)]} \overrightarrow{e_\lambda} = \frac{B_i(\lambda)}{\langle \overrightarrow{R(\lambda)}, \overrightarrow{B(\lambda)} \rangle} \overrightarrow{e_\lambda} = \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \overrightarrow{e_\lambda} \quad (4.29)$$

Cette méthode est nommée séquentielle car elle consiste à appliquer la méthode de résolution d'une contrainte sur chaque contrainte une à une en utilisant la formulation (4.29). Ce processus est répété jusqu'à ce que la précision désirée soit obtenue. Nous considérons qu'une application du processus consiste à résoudre les N contraintes, l'une après l'autre. L'algorithme 3 que nous avons mis au point pour cette méthode est consultable dans l'annexe D et la figure 4.63 illustre son déroulement pas à pas sur une courbe B-Spline de degré 3 définie par 5 points de contrôle. Les 5 contraintes de localisation utilisées pour cet exemple correspondent à l'influence *naturelle* des pôles. Les colonnes représentent les itérations et les lignes les contraintes G_0 à G_4 à satisfaire.

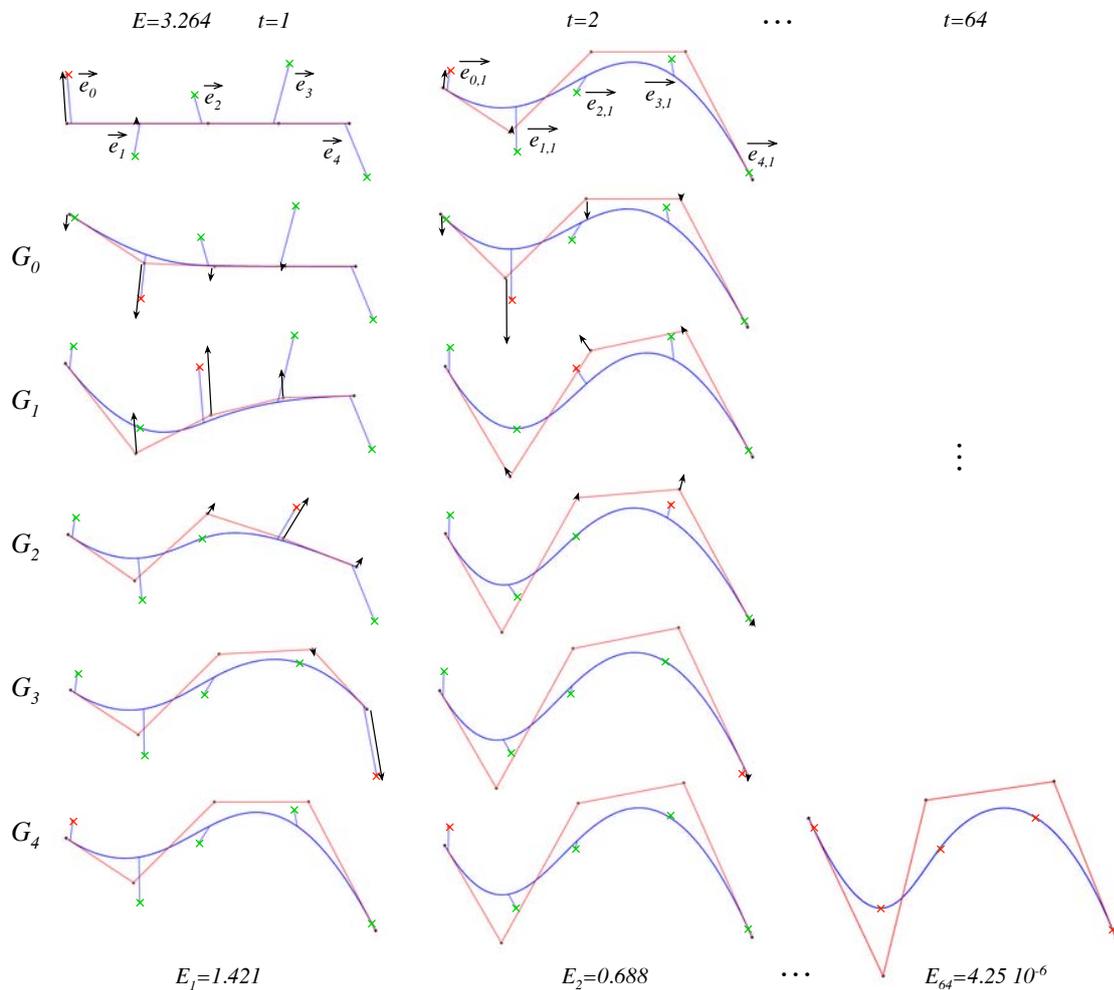


Figure 4.63. Déroulement pas à pas de la méthode séquentielle sur une courbe B-Spline de degré 3 définie par 5 pôles (16 ms sur un P4 - 2.4 Ghz - 1.00 Go RAM)

L'un des points importants de cette méthode que l'on peut constater sur la figure précédente, consiste à bien comprendre que chaque contrainte est parfaitement satisfaite l'une après

l'autre. Ainsi lorsque la contrainte G_λ est satisfaite, une nouvelle erreur est ajoutée à toutes les autres contraintes qui sont dans son influence. C'est pourquoi nous faisons intervenir plusieurs nouvelles notions attachées au concept d'erreur locale telles que :

- Soit $\overrightarrow{\varepsilon_{\lambda,k,t}}$ le nouveau vecteur erreur attaché à G_λ après la satisfaction de G_k durant l'itération t . Il est à noter que $\overrightarrow{\varepsilon_{\lambda,\lambda,t}} = \overrightarrow{0}$;
- Soit $\varepsilon_{\lambda,t} = \varepsilon_{\lambda,\lambda-1,t}$ la dernière erreur connue attachée à G_λ avant sa satisfaction durant l'itération t . C'est donc l'erreur à prendre en compte pour satisfaire la contrainte G_λ pendant l'itération t .

Une relation importante peut être déduite des deux dernières définitions :

Lorsque l'itération t est terminée, c'est-à-dire que toutes les contraintes ont été traitées, on a $\overrightarrow{e_{\lambda,t}} = \overrightarrow{\varepsilon_{\lambda,N-1,t}}$. À ce stade, le vecteur erreur attaché à chaque contrainte G_λ est $\overrightarrow{e_{\lambda,t}}$. Afin de gérer la première erreur du processus, rappelons que $\overrightarrow{e_{\lambda,0}} = \overrightarrow{e_\lambda}$.

Convention de notation : Dans l'expression de la correction de la contrainte k , il apparaît naturellement les erreurs liées à la correction de la contrainte $k - 1$. En d'autres termes, $\overrightarrow{\varepsilon_{\lambda,k,t}}$ dépend de $\overrightarrow{\varepsilon_{\lambda,k-1,t}}$. Ceci pose un problème de notation quand $k = 0$ (correction de la première contrainte). Pour généraliser les notations, on notera $\overrightarrow{\varepsilon_{\lambda,-1,t}} = \overrightarrow{\varepsilon_{\lambda,N-1,t-1}} = \overrightarrow{e_{\lambda,t-1}}$. Ce qui revient à prendre en compte, quand $k = 0$, l'erreur sur la contrainte λ à l'étape précédente ($t - 1$).

Avec ces considérations et en tenant compte de la convention de notation précédente, il est possible d'exprimer, pour l'itération t , le déplacement à appliquer à P_i pour satisfaire G_λ . Ce déplacement est noté $\overrightarrow{m_{\lambda,t}(i)}$ et se formalise, à partir de (4.29) tel que :

$$\overrightarrow{m_{\lambda,t}(i)} = \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \overrightarrow{\varepsilon_{\lambda,\lambda-1,t}} = \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \overrightarrow{\varepsilon_{\lambda,t}} \quad \text{avec } t > 0 \quad (4.30)$$

En pratique, la méthode converge parfaitement lorsqu'il existe une déformation de la courbe permettant de satisfaire exactement toutes les contraintes (spatiales, paramétriques et d'influence). La vitesse de convergence est d'autant plus rapide que l'influence des contraintes est éloignée. Si les contraintes n'ont aucune influence les unes sur les autres, le processus déforme la courbe en une itération et satisfait toutes les contraintes. La méthode séquentielle converge sensiblement plus vite que la méthode globale. Cependant, contrairement à cette dernière, elle diverge dans le cas d'un système surcontraint. Il est donc important de prévoir dans l'algorithme un procédé de détection permettant d'arrêter le processus le cas échéant. D'un point de vue théorique, la convergence de cette méthode reste à démontrer. Néanmoins, dans le cas de sa convergence, elle peut s'exprimer sous la même forme algébrique que la méthode globale, c'est ce que nous présentons dans le paragraphe suivant.

4.3.6.1.5 Expression algébrique de la méthode séquentielle

Afin d'exprimer la méthode séquentielle sous sa forme algébrique, considérons en premier

lieu, le déplacement total appliqué au point de contrôle P_i à la fin de l'itération t telle que :

$$\overrightarrow{m}_t(i) = \sum_{\lambda=0}^{N-1} \overrightarrow{m}_{\lambda,t}(i) \quad (4.31)$$

La nouvelle erreur attachée à G_λ après la satisfaction de G_k est la dernière erreur connue de G_λ moins l'erreur ajoutée par la correction de G_k :

$$\overrightarrow{\varepsilon}_{\lambda,k,t} = \overrightarrow{\varepsilon}_{\lambda,k-1,t} - \sum_{i=0}^{n-1} R_i(\lambda) \overrightarrow{m}_{k,t}(i) = \overrightarrow{\varepsilon}_{\lambda,k-1,t} - \frac{\sum_{i=0}^{n-1} [R_i(\lambda) B_i(k)]}{D_{k,k}} \overrightarrow{\varepsilon}_{k,t} \text{ avec } k < \lambda \text{ et } \lambda \geq 1$$

mais $\sum_{i=0}^{n-1} [R_i(\lambda) B_i(k)] = \langle \overrightarrow{R}(\lambda), \overrightarrow{B}(k) \rangle = D_{\lambda,k}$, donc :

$$\overrightarrow{\varepsilon}_{\lambda,k,t} = \overrightarrow{\varepsilon}_{\lambda,k-1,t} - \frac{D_{\lambda,k}}{D_{k,k}} \overrightarrow{\varepsilon}_{k,t} \quad (4.32)$$

Proposition 4.3.8 : La dernière erreur connue de G_λ avant sa satisfaction durant l'itération t est notée :

$$\overrightarrow{\varepsilon}_{\lambda,t} = \overrightarrow{e}_{\lambda,t-1} - \sum_{k=0}^{\lambda-1} \tilde{C}_{\lambda,k} \overrightarrow{e}_{k,t-1} \quad (4.33)$$

Où \tilde{C} est la matrice triangulaire inférieure de taille $N \times N$ indexée de 0 à $N-1$ telle que :

$$\tilde{C}_{\lambda,k} = \begin{cases} \frac{D_{\lambda,k}}{D_{k,k}} - \sum_{l=k+1}^{\lambda-1} \frac{D_{\lambda,l}}{D_{l,l}} \tilde{C}_{l,k} & \text{si } \lambda \in [1, N-1] \text{ et } k \in [0, \lambda-1] \\ 0 & \text{sinon} \end{cases}$$

$$\tilde{C} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{D_{1,0}}{D_{0,0}} & 0 & 0 & \cdots & 0 & 0 \\ \left(\frac{D_{2,0}}{D_{0,0}} - \frac{D_{2,1}}{D_{1,1}} \tilde{C}_{1,0} \right) & \frac{D_{2,1}}{D_{1,1}} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \frac{D_{N-2,N-3}}{D_{N-3,N-3}} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \frac{D_{N-1,N-2}}{D_{N-2,N-2}} & 0 \end{pmatrix}$$

Cette proposition est démontrée par récurrence sur λ par la preuve E.2 disponible en annexes.

L'expression (4.33) peut se simplifier en utilisant une nouvelle matrice $N \times N$ telle que

$C = I - \tilde{C}$. On a alors :

$$\overrightarrow{\varepsilon_{\lambda,t}} = \sum_{k=0}^{\lambda} C_{\lambda,k} \overrightarrow{e_{k,t-1}} \quad (4.34)$$

Posons F la matrice $N \times N$ diagonale définie par $F_{i,i} = D_{i,i}$.

Proposition 4.3.9 : L'erreur après t itérations attachée à G_{λ} peut être calculée en fonction de l'erreur de cette même contrainte, connue après $t - 1$ itérations telle que :

$$\overrightarrow{e_{\lambda,t}} = \overrightarrow{\varepsilon_{\lambda,N-1,t}} = \overrightarrow{e_{\lambda,t-1}} - \sum_{k=0}^{N-1} \tilde{S}_{\lambda,k} \overrightarrow{e_{k,t-1}} \quad (4.35)$$

Où \tilde{S} est la matrice $N \times N$ définie par : $\tilde{S} = DF^{-1}C$

Cette proposition est démontrée dans la preuve E.3 disponible en annexes.

L'expression (4.35) peut une fois encore se simplifier en utilisant une nouvelle matrice $N \times N$ $S = I - \tilde{S}$ telle que :

$$\overrightarrow{e_{\lambda,t}} = \sum_{k=0}^{N-1} S_{\lambda,k} \overrightarrow{e_{k,t-1}} \quad (4.36)$$

Proposition 4.3.10 : Prouvons que le vecteur erreur à la fin de l'itération t peut être calculé en fonction de t :

$$\overrightarrow{e_{\lambda,t}} = \sum_{k=0}^{N-1} [S^t]_{\lambda,k} \overrightarrow{e_k} \quad (4.37)$$

Preuve :

Soit $t > 0$. En utilisant (4.36), la preuve est faite par récurrence sur t .

$$- \text{Cas de } t = 1 : \overrightarrow{e_{\lambda,1}} = \sum_{k=0}^{N-1} S_{\lambda,k} \overrightarrow{e_{k,0}} = \sum_{k=0}^{N-1} S_{\lambda,k} \overrightarrow{e_k}$$

- Si l'hypothèse (4.37) est vraie pour t , l'est-elle encore pour $t + 1$?

$$\begin{aligned} \overrightarrow{e_{\lambda,t+1}} &= \sum_{k=0}^{N-1} S_{\lambda,k} \overrightarrow{e_{k,t}} \text{ utilisons l'hypothèse (4.37)} \\ &= \sum_{l=0}^{N-1} S_{\lambda,l} \sum_{k=0}^{N-1} [S^t]_{l,k} \overrightarrow{e_k} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} S_{\lambda,l} [S^t]_{l,k} \overrightarrow{e_k} = \sum_{k=0}^{N-1} [S^{t+1}]_{\lambda,k} \overrightarrow{e_k} \end{aligned}$$

Ce qui permet de conclure.

□

En utilisant (4.37), l'équation (4.34) devient :

$$\overrightarrow{\varepsilon}_{\lambda,t} = \sum_{l=0}^{\lambda} C_{\lambda,l} \sum_{k=0}^{N-1} [S^{t-1}]_{l,k} \overrightarrow{e}_k = \sum_{k=0}^{N-1} \sum_{l=0}^{\lambda} C_{\lambda,l} [S^{t-1}]_{l,k} \overrightarrow{e}_k \quad (4.38)$$

Or la matrice C est une matrice triangulaire inférieure, c'est pourquoi il est possible de réécrire (4.38) de la manière suivante :

$$\overrightarrow{\varepsilon}_{\lambda,t} = \sum_{k=0}^{N-1} [CS^{t-1}]_{\lambda,k} \overrightarrow{e}_k \quad (4.39)$$

Ainsi, en utilisant (4.30), (4.31) et (4.39), le déplacement total appliqué à P_i à la fin de l'itération t s'écrit :

$$\overrightarrow{m}_t(i) = \sum_{\lambda=0}^{N-1} \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \sum_{k=0}^{N-1} [CS^{t-1}]_{\lambda,k} \overrightarrow{e}_k \quad (4.40)$$

Le déplacement total appliqué à chaque point de contrôle P_i depuis la première itération jusqu'à l'itération t est notée $\overrightarrow{m}_{1 \rightarrow t}(i)$ et s'écrit tel que :

$$\begin{aligned} \overrightarrow{m}_{1 \rightarrow t}(i) &= \sum_{l=1}^t \overrightarrow{m}_l(i) = \sum_{l=1}^t \sum_{\lambda=0}^{N-1} \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \sum_{k=0}^{N-1} [CS^{l-1}]_{\lambda,k} \overrightarrow{e}_k \\ &= \sum_{\lambda=0}^{N-1} \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \sum_{k=0}^{N-1} \left[\sum_{l=1}^t CS^{l-1} \right]_{\lambda,k} \overrightarrow{e}_k \\ &= \sum_{\lambda=0}^{N-1} \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \sum_{k=0}^{N-1} \left[C \sum_{l=1}^t S^{l-1} \right]_{\lambda,k} \overrightarrow{e}_k \end{aligned}$$

Ainsi :

$$m_i(t) = \lim_{t \rightarrow +\infty} \overrightarrow{m}_{1 \rightarrow t}(i) = \lim_{t \rightarrow +\infty} \sum_{\lambda=0}^{N-1} \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \sum_{k=0}^{N-1} \left[C \sum_{l=1}^t S^{l-1} \right]_{\lambda,k} \overrightarrow{e}_k \quad (4.41)$$

Proposition 4.3.11 : Si $\lim_{t \rightarrow +\infty} \sum_{l=1}^t S^{l-1}$ existe et est finie, alors la méthode converge vers la solution correcte (satisfaction de toutes les contraintes).

Dans ce cas, on peut alors écrire :

$$m_i(t) = \lim_{t \rightarrow +\infty} \overrightarrow{m}_{1 \rightarrow t}(i) = \sum_{\lambda=0}^{N-1} \frac{B_i(\lambda)}{D_{\lambda,\lambda}} \sum_{k=0}^{N-1} \left[C \lim_{t \rightarrow +\infty} \sum_{l=1}^t S^{l-1} \right]_{\lambda,k} \overrightarrow{e}_k$$

Comme pour la méthode globale, il faut montrer que le rayon spectral de S est strictement inférieur à 1. Cette preuve n'est pas aboutie à ce jour et reste une perspective importante de la suite des travaux attachés à cette méthode.

Dans le cas où cette limite existe et est finie, le corollaire suivant peut être exprimé :

Corollaire 4.3.1 : Puisque $S = I - \tilde{S}$, si S est une matrice convergente alors la matrice \tilde{S} est inversible et :

$$\tilde{S}^{-1} = (I - S)^{-1} = \lim_{T \rightarrow +\infty} \sum_{t=0}^T S^t \quad (4.42)$$

Proposition 4.3.12 : La formulation algébrique du déplacement $\overrightarrow{m(i)}$ à appliquer à chaque point de contrôle P_i pour satisfaire parfaitement l'ensemble des contraintes ponctuelles de passage est :

$$\overrightarrow{m(i)} = \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} [D^{-1}]_{\lambda,k} \vec{e}_k \quad (4.43)$$

Preuve :

Grâce à (4.42), on peut dire que $\lim_{T \rightarrow +\infty} \sum_{t=1}^T S^{t-1} = \tilde{S}^{-1}$.

Or puisque \tilde{S} est inversible, son déterminant est différent de 0. On sait que $\tilde{S} = DF^{-1}C$, donc la somme des déterminants de D , F^{-1} et C est différente de 0. On peut ainsi dire que D , F^{-1} et C sont inversibles, ce qui donne : $\tilde{S}^{-1} = C^{-1}FD^{-1}$. La formulation (4.41) devient alors :

$$\begin{aligned} \overrightarrow{m(i)} &= \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} \frac{[C\tilde{S}^{-1}]_{\lambda,k}}{D_{\lambda,\lambda}} \vec{e}_k = \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} \frac{[CC^{-1}FD^{-1}]_{\lambda,k}}{D_{\lambda,\lambda}} \vec{e}_k \\ &= \sum_{\lambda=0}^{N-1} B_i(\lambda) \sum_{k=0}^{N-1} [D^{-1}]_{\lambda,k} \vec{e}_k \end{aligned}$$

□

Cette formulation est la même que pour la méthode globale (4.28). Ainsi, s'il existe une solution satisfaisant toutes les contraintes ponctuelles, les deux méthodes convergent vers le même résultat.

Nous proposons donc deux méthodes itératives appliquées aux courbes, permettant de résoudre le même problème de contraintes ponctuelles de passage par un principe de déformation. La suite de ce chapitre fait un bref comparatif de ces deux méthodes avant de passer à leurs applications sur les surfaces.

4.3.6.1.6 Comparaison des 2 méthodes

La formulation algébrique des deux méthodes proposées nécessite l'inversion d'une matrice $N \times N$ (N étant le nombre de contraintes) qui peut être un calcul instable selon les méthodes

employées. L'inversion d'une matrice peut en outre demander un temps de calcul important pour un nombre de contraintes élevé.

C'est pourquoi, nous optons pour l'utilisation de l'une des méthodes itératives proposées qui donne la possibilité à l'utilisateur de stopper le processus dès que la précision désirée est atteinte. De plus une contrainte dite *lâche* peut être considérée comme ayant une précision moins importante que des contraintes plus *dures*, ce qui permet une plus grande souplesse dans les résultats obtenus, notion essentielle en modélisation déclarative.

Le tableau suivant regroupe les résultats numériques¹² obtenus pour les exemples¹³ des figures 4.60 et 4.63 :

Méthode	Temps (ms)	Erreur finale (précision)	Nb d'itérations
Globale	140	4.8810^{-6}	1852
Séquentielle	16	4.2510^{-6}	43
Algébrique	< 1	10^{-13}	

TAB. 4.1. Résultats numériques de la satisfaction de 5 contraintes ponctuelles de passage ayant une influence *naturelle*

Il s'avère qu'en pratique, la vitesse de convergence de la méthode globale peut être nettement améliorée en appliquant un coefficient aux déplacements (4.21) de chaque pôle de la courbe. Ce coefficient que l'on qualifie d'optimal varie pour chaque contrainte. Ainsi une perspective de travail est d'accélérer la convergence de la méthode globale en trouvant ce coefficient optimal.

De par leur approche géométrique, ces deux méthodes peuvent être facilement étendues aux surfaces. La plus importante différence réside dans le fait que pour les courbes, $\overrightarrow{R(\lambda)}$ et $\overrightarrow{B(\lambda)}$ sont des vecteurs alors que ce sont des matrices dans le cas des surfaces. Dans la partie suivante, nous donnons donc les formules et les notations se rapportant au cas des surfaces. En pratique les résultats sont parfaitement stables et cohérents, cependant les démonstrations étant beaucoup plus délicates que pour les courbes, nous n'avons pas souhaité les développer durant cette thèse. Elles restent des perspectives se rattachant à l'évolution des méthodes.

4.3.6.2 Application aux surfaces

4.3.6.2.1 Notations

Dans cette partie nous ne définissons ou ne redéfinissons que les éléments nécessaires et spécifiques aux surfaces.

- Soit N contraintes G_λ ($\lambda \in [0, N - 1]$) chacune définie pour des paramètres $(u_\lambda, v_\lambda) \in [U_0, U_{p+n}] \times [V_0, V_{q+m}]$ et dont la fonction d'influence (ou contrainte de localisation)

¹²sur un P4 - 2.4 Ghz - 1.00 Go RAM

¹³une courbe B-Spline de degré 5 définie par 5 points de contrôle

est $f_\lambda(i, j)$. Chaque G_λ doit déformer \mathcal{S} en $\hat{\mathcal{S}}$ afin d'obtenir $\hat{\mathcal{S}}(u_\lambda, v_\lambda) = \hat{M}_\lambda$. Il est à remarquer que $N \leq (n \times m)$ est une condition nécessaire pour obtenir une solution ;

– Considérons N matrices $R(\lambda)$ de taille $n \times m$:

$$R(\lambda) = \begin{pmatrix} R_{0,0}(u_\lambda, v_\lambda) & R_{0,1}(u_\lambda, v_\lambda) & \cdots & R_{0,m-1}(u_\lambda, v_\lambda) \\ R_{1,0}(u_\lambda, v_\lambda) & R_{1,1}(u_\lambda, v_\lambda) & \cdots & R_{1,m-1}(u_\lambda, v_\lambda) \\ \vdots & \vdots & \ddots & \vdots \\ R_{n-1,0}(u_\lambda, v_\lambda) & R_{n-1,1}(u_\lambda, v_\lambda) & \cdots & R_{n-1,m-1}(u_\lambda, v_\lambda) \end{pmatrix}$$

$R(\lambda)$ est ainsi défini à partir des fonctions B-Spline attachées à la contrainte G_λ ;

– Définissons N vecteurs $\tilde{B}(\lambda)$ de taille $n \times m$, où chaque $\tilde{B}_{i,j}(\lambda) = f_\lambda(i, j)$ est l'influence¹⁴ de G_λ sur le point de contrôle $P_{i,j}$ tel que :

$$\tilde{B}(\lambda) = \begin{pmatrix} f_\lambda(0,0) & f_\lambda(0,1) & \cdots & f_\lambda(0,m-1) \\ f_\lambda(1,0) & f_\lambda(1,1) & \cdots & f_\lambda(1,m-1) \\ \vdots & \vdots & \ddots & \vdots \\ f_\lambda(n-1,0) & f_\lambda(n-1,1) & \cdots & f_\lambda(n-1,m-1) \end{pmatrix}$$

– Soit alors $B(\lambda)$ tel que : $B_{i,j}(\lambda) = \frac{|\tilde{B}_{i,j}(\lambda)|}{\|\tilde{B}(\lambda)\|_1}$, où $\|\cdot\|_1$ est la norme matricielle définie

$$\text{par : } \|A\|_1 = \max_{k=1,\dots,n} \sum_{l=1}^m |a_{k,l}|.$$

– Soit D la matrice $N \times N$ définie par :

$$D_{k,l} = \langle R(k), B(l) \rangle = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [R_{i,j}(k) B_{i,j}(l)]$$

$$D = \begin{pmatrix} \langle R(0), B(0) \rangle & \langle R(0), B(1) \rangle & \cdots & \langle R(0), B(N-1) \rangle \\ \langle R(1), B(0) \rangle & \langle R(1), B(1) \rangle & \cdots & \langle R(1), B(N-1) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle R(N-1), B(0) \rangle & \langle R(N-1), B(1) \rangle & \cdots & \langle R(N-1), B(N-1) \rangle \end{pmatrix}$$

Où $\langle \cdot, \cdot \rangle$ est le produit scalaire défini dans l'espace matriciel.

Remarque 4.3.9 – $0 < D_{k,l} \leq 1$ puisque $R(k)$ et $B(l)$ ont des coefficients positifs ou nuls, inférieurs ou égaux à 1.

4.3.6.2.2 Méthode 1 : Approche globale

La méthode globale appliquée aux surfaces est sensiblement la même que celle appliquée sur les courbes. Le déplacement appliqué à chaque point de contrôle $P_{i,j}$ lors de chaque itération t est :

$$\overrightarrow{m_t(i, j)} = \frac{1}{N} \sum_{\lambda=0}^{N-1} B_{i,j}(\lambda) \overrightarrow{e_{\lambda,t-1}} \text{ avec } t > 0 \quad (4.44)$$

¹⁴ $\tilde{B}_{i,j}(\lambda) = R_{i,j}(\lambda)$ correspond donc à la déformation naturelle ou $B(\lambda) = R(\lambda)$

La figure 4.64 illustre le résultat de l'application de cette méthode sur une surface B-Spline de degré 3×3 définie par 30×20 pôles. L'influence des 10 contraintes à satisfaire est définie par l'influence naturelle des pôles filtrée par h de type gaussien et de rayon=14.

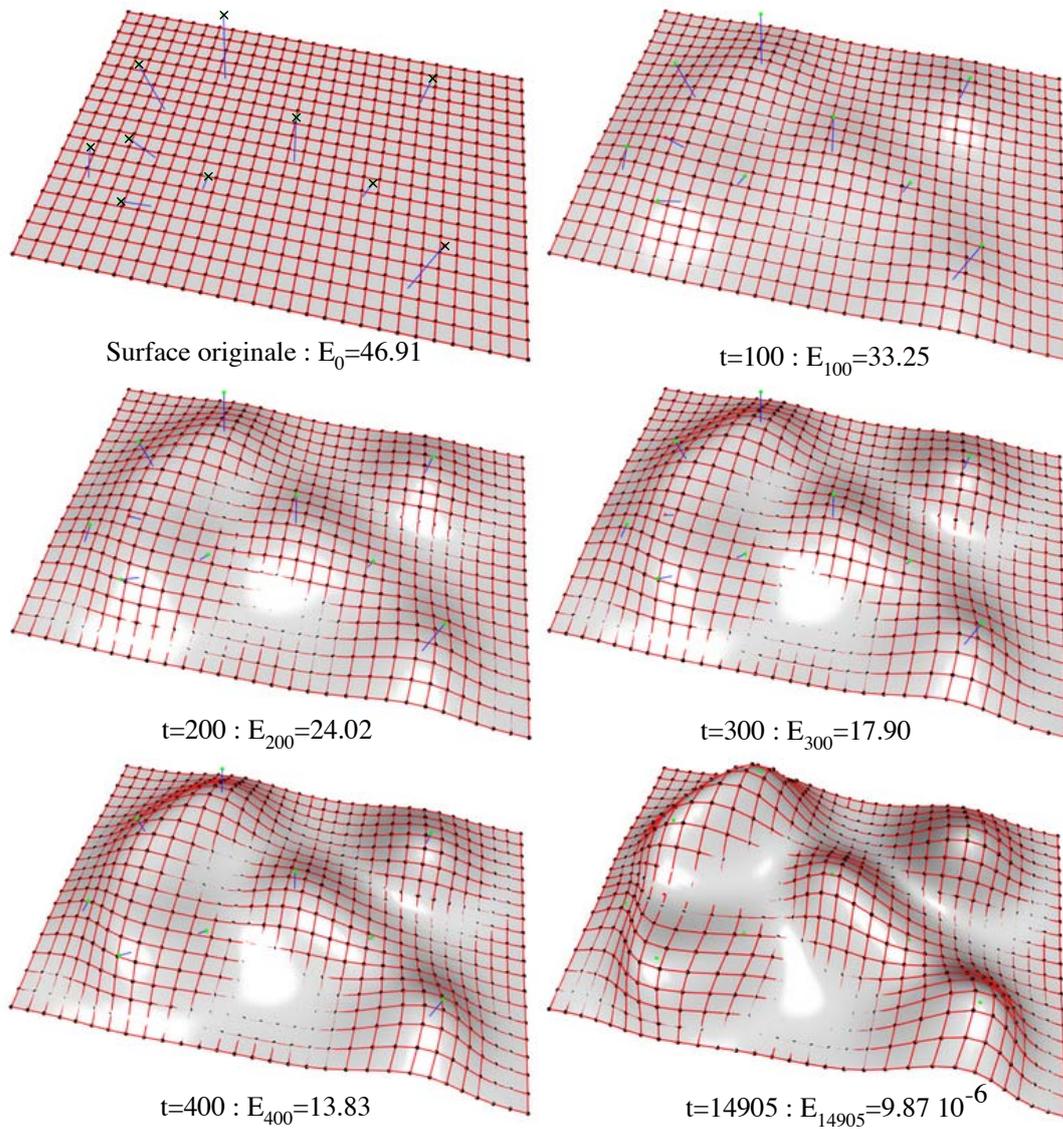


Figure 4.64. Déroulement pas à pas de la méthode globale sur une surface (16.04 s sur un P4 - 2.4 Ghz - 1.00 Go RAM)

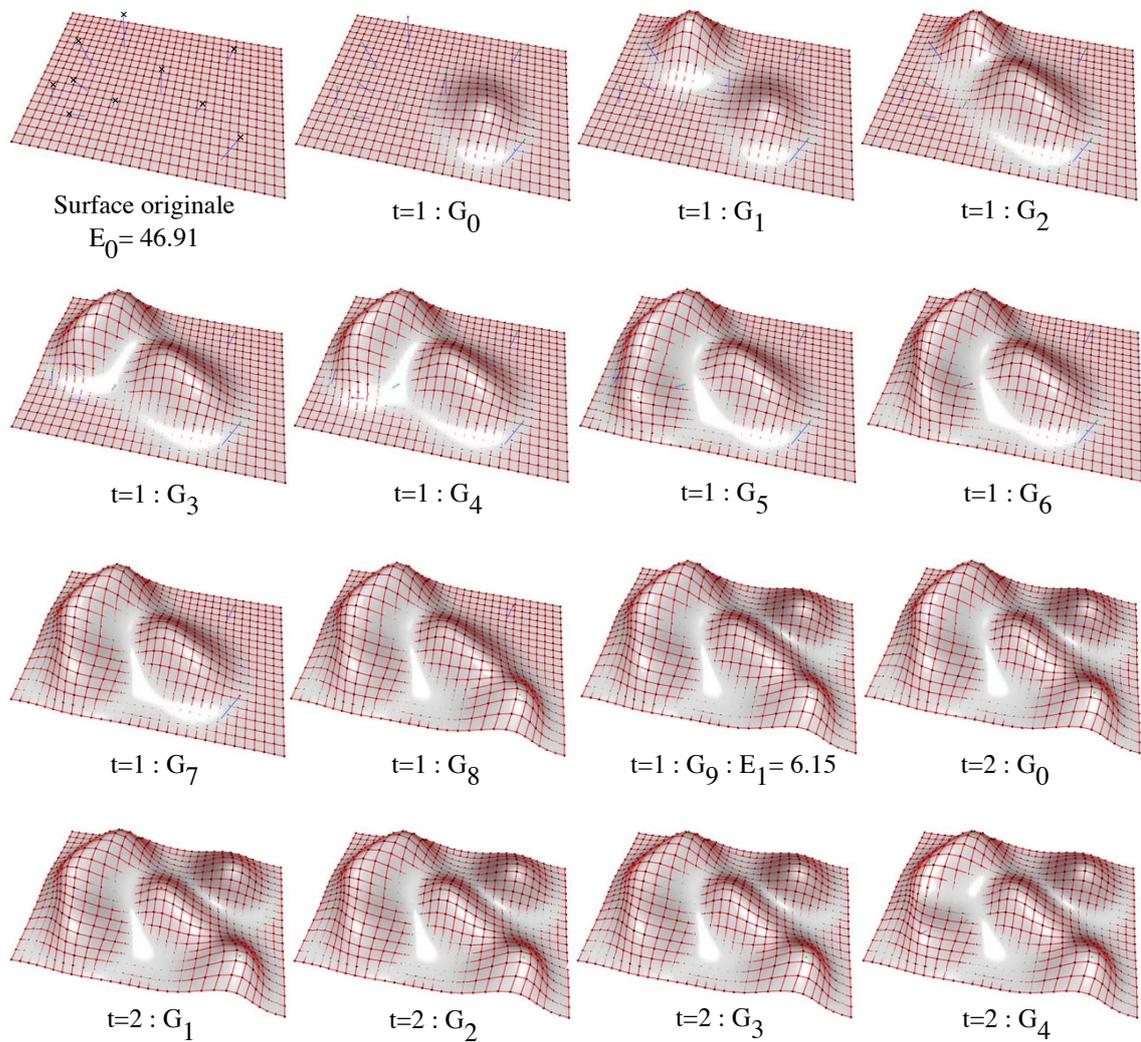
4.3.6.2.3 Méthode 2 : Approche séquentielle

L'approche séquentielle appliquée aux surfaces suit les mêmes principes que celle appliquée aux courbes. Il suffit de suivre les résultats de la partie 4.3.5.2 selon le principe vu au paragraphe 4.3.6.1.4. Ainsi, on obtient le déplacement à appliquer à chaque point de contrôle

$P_{i,j}$ lors de chaque itération t tel que :

$$\overrightarrow{m_{\lambda,t}(i,j)} = \frac{B_{i,j}(\lambda)}{D_{\lambda,\lambda}} \overrightarrow{\varepsilon_{\lambda,\lambda-1,t}} = \frac{B_{i,j}(\lambda)}{D_{\lambda,\lambda}} \overrightarrow{\varepsilon_{\lambda,t}} \text{ avec } t > 0 \quad (4.45)$$

La figure 4.65 illustre le résultat de l'application de cette méthode sur une surface B-Spline de degré 3×3 définie par 30×20 pôles. L'influence des 10 contraintes à satisfaire est définie par l'influence naturelle des pôles filtrée par h de type gaussien et de rayon=14.



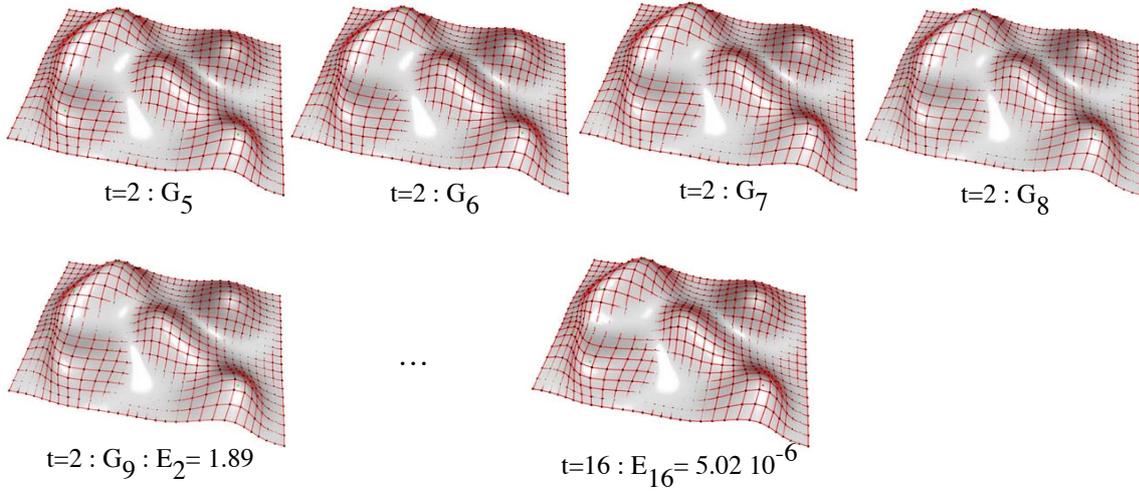


Figure 4.65. Déroulement pas à pas de la méthode séquentielle sur une surface (219 ms sur un P4 - 2.4 Ghz - 1.00 Go RAM)

4.3.6.2.4 Formulation algébrique et liens entre les méthodes

La formulation algébrique du déplacement à appliquer à chaque point de contrôle $P_{i,j}$ de \mathcal{S} afin d'obtenir $\hat{\mathcal{S}}$ vérifiant $\hat{\mathcal{S}}(u_\lambda, v_\lambda) = \hat{M}_\lambda \quad \forall \lambda \in [0, N - 1]$ est $\overrightarrow{m(i, j)}$ défini tel que :

$$\overrightarrow{m(i, j)} = \sum_{\lambda=0}^{N-1} B_{i,j}(\lambda) \sum_{k=0}^{N-1} [D^{-1}]_{\lambda,k} \vec{e}_k \quad (4.46)$$

La figure 4.66 illustre le résultat de l'application de cette formule sur une surface B-Spline de degré 3×3 définie par 30×20 pôles. L'influence des 10 contraintes à satisfaire est définie par l'influence naturelle des pôles filtrée par h de type gaussien et de rayon=14.

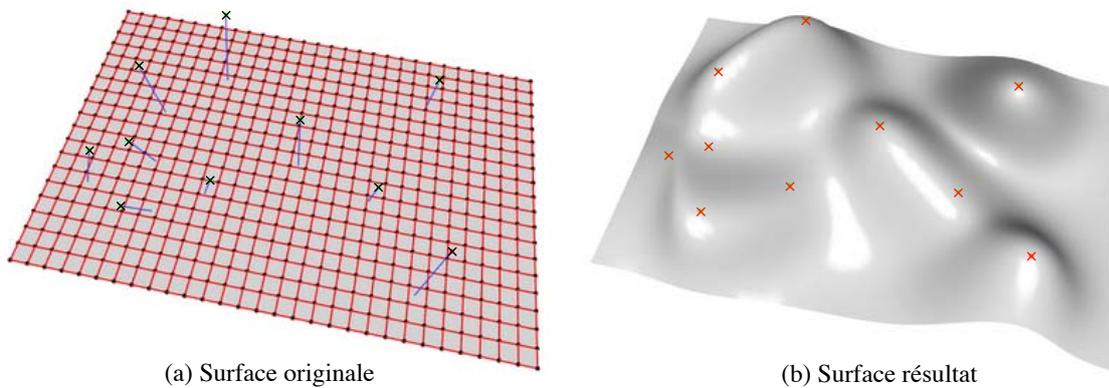


Figure 4.66. Résultat de la formulation algébrique sur une surface (235 ms sur un P4 - 2.4 Ghz - 1.00 Go RAM)

Le tableau suivant regroupe les résultats numériques¹⁵ obtenus pour l'exemple¹⁶ des figures 4.64 et 4.65 :

Méthode	Temps (s)	Erreur finale (précision)	Nb d'itérations
Globale	16.406	9.9710^{-6}	14905
Séquentielle	0.219	5.0210^{-6}	16
Algébrique	0.235	10^{-13}	

TAB. 4.2. Résultats numériques de la satisfaction de 10 contraintes ponctuelles de passage sur une surface

Bien qu'il existe sur cet exemple plusieurs cas d'influences réciproques entre contraintes, il s'avère que la résolution itérative séquentielle est un peu plus rapide que la formulation algébrique pour une erreur finale de l'ordre de 10^{-6} . La remarque concernant la vitesse de convergence de la méthode globale est ici bien plus flagrante que sur les courbes, puisqu'elle est près de 75 fois plus lente, en temps, que la méthode séquentielle avec 930 fois plus d'itérations.

Voyons maintenant, dans la partie suivante, comment intégrer ces méthodes de résolution et de déformations sous contraintes dans notre processus déclaratif en liant contraintes *lâches* et *dures*.

4.3.7 Gestion de certaines contraintes «lâches» comme cas particuliers de contraintes «dures»

Une contrainte dite «lâche» est, nous le rappelons, une tendance que veut voir apparaître l'utilisateur dans une zone de la surface. Il peut cependant préciser cette contrainte lâche en donnant une contrainte de passage comme par exemple : «*Z1 est bombée et passe par le point A*». Avec la méthode actuelle, nous pouvons arriver à des résultats ne correspondant pas forcément à la première idée de l'utilisateur.

La figure 4.67 présente d'ailleurs sur une courbe des résultats possibles lorsque la contrainte lâche est d'abord appliquée (Fig. 4.67.b, 4.67.c) sans directement satisfaire la contrainte de passage. On obtient alors des déformations faisant apparaître un pic (Fig. 4.67.d) ou un creux (Fig. 4.67.e) selon que la déformation bombée dépasse ou non la contrainte de passage à satisfaire. Ces résultats ne sont pas faux puisque de façon globale la zone reste bombée et la contrainte de passage est respectée, mais cela ne correspond pas forcément à l'idée que s'en faisait l'utilisateur. Un résultat qui semble être plus naturel est présenté sur la figure 4.67.f où la déformation lâche est calculée de manière à respecter la contrainte de passage.

¹⁵sur un P4 - 2.4 Ghz - 1.00 Go RAM

¹⁶une surface B-Spline de degré 3×3 définie par 30×20 points de contrôle. L'influence des 10 contraintes à satisfaire est définie par l'influence naturelle des pôles filtrée par h de type gaussien et de rayon=14

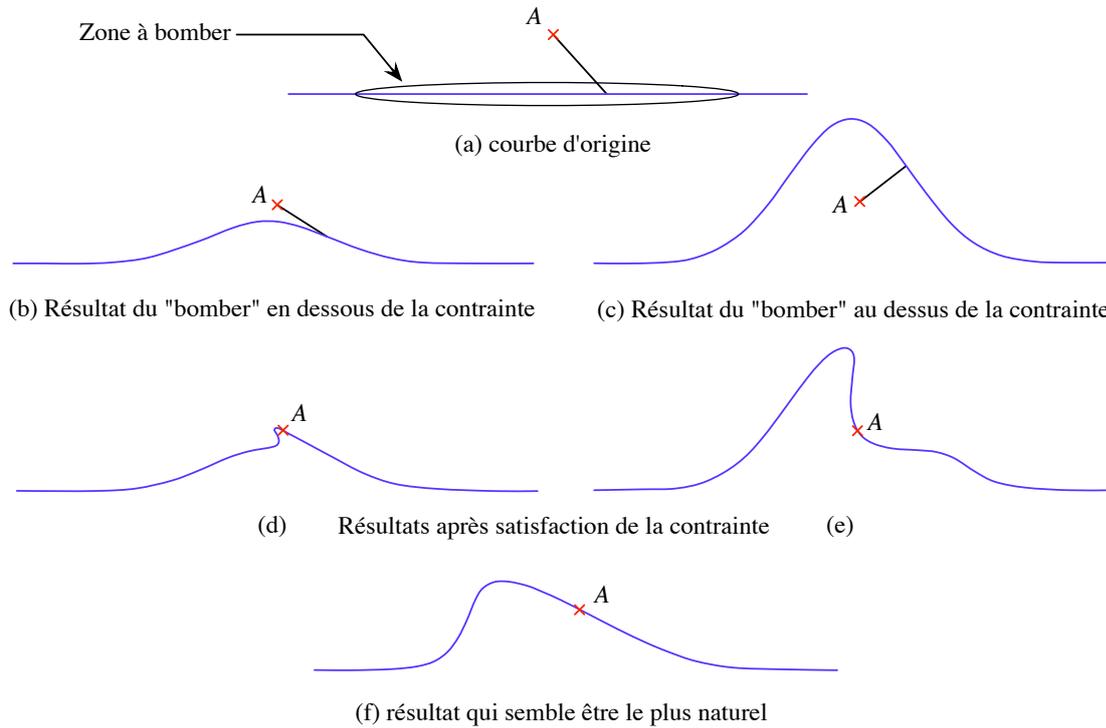


Figure 4.67. Exemples de déformation : «La zone est bombée et passe par le point A»

Ainsi, bien que l'utilisateur précise la tendance de la forme de la surface sur une zone, ce type de contrainte ne doit pas être vue comme une contrainte lâche, mais comme une contrainte dure dont on précise en plus une contrainte de localité.

La détection de ces cas doit être faite au niveau du processus de traduction sémantique. Sur cet exemple, la méthode consiste à définir une contrainte G_λ telle que :

- Sa contrainte de localité, $B(\lambda)$, correspond à la grille de distance attachée aux pôles de $Z1$;
- Sa contrainte paramétrique est simplement un couple (u_λ, v_λ) appartenant à $Z1$;
- Sa contrainte spatiale est le point A en question.

Cette méthode est aussi applicable à des déformations de type *creuser*.

Un autre cas remarquable est celui de la proximité, comme par exemple « $Z1$ est bombée et passe près du point A ». Ainsi, il n'y a aucune obligation de satisfaction parfaite de la contrainte de passage. On peut donc définir le terme de contrainte *floue*, comme une contrainte qui peut être satisfaite à une erreur donnée près. Une façon de formaliser cette notion est d'attacher à chaque contrainte G_λ une erreur acceptable pour laquelle on peut considérer la contrainte satisfaite.

Si l'on note ε_λ l'erreur acceptable attachée à G_λ , alors G_λ est considérée comme parfaitement satisfaite si $\|\mathcal{S}(u_\lambda, v_\lambda)\hat{M}_\lambda\|_2 < \varepsilon_\lambda$. La figure 4.68 schématise quelques exemples de cette notion sur la base d'une courbe.

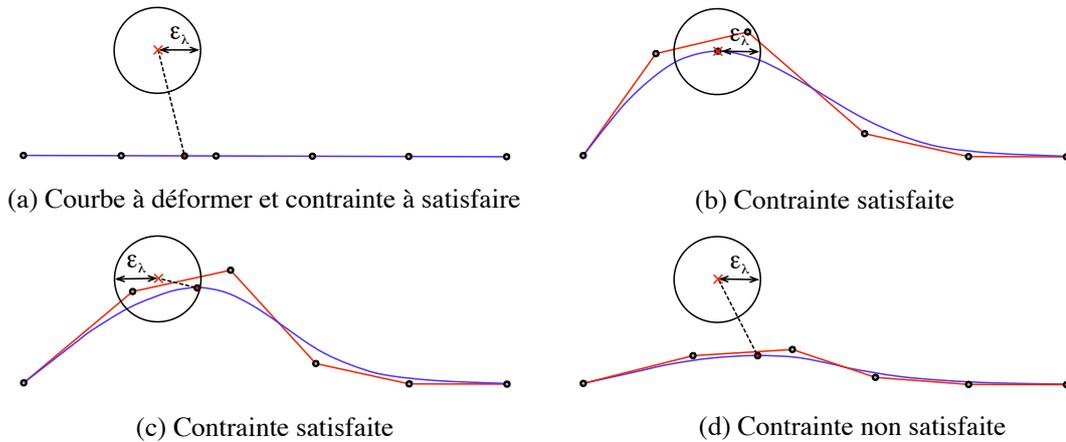


Figure 4.68. Notion de contraintes *floues* pour la résolution de contraintes de proximité

Remarque 4.3.10 – Cette nouvelle interprétation d'une contrainte peut être intégrée dans les processus de résolution itératifs que l'on a présentés dans la partie 4.3.6. En effet, il suffit de considérer l'erreur que présente la contrainte G_λ comme nulle si elle est inférieure à ε_λ dans les conditions d'arrêt de techniques. Les algorithmes de résolution doivent aussi en tenir compte lors des calculs d'erreurs.

À partir de l'ensemble de ces outils qui permettent de traiter des contraintes lâches et des contraintes dures, nous présentons dans la partie suivante le principe de fonctionnement du moteur de résolution/construction utilisé.

4.3.8 Fonctionnement global du moteur de résolution/construction

Le moteur de résolution/construction a pour but de générer une instance représentative d'un modèle patron donné.

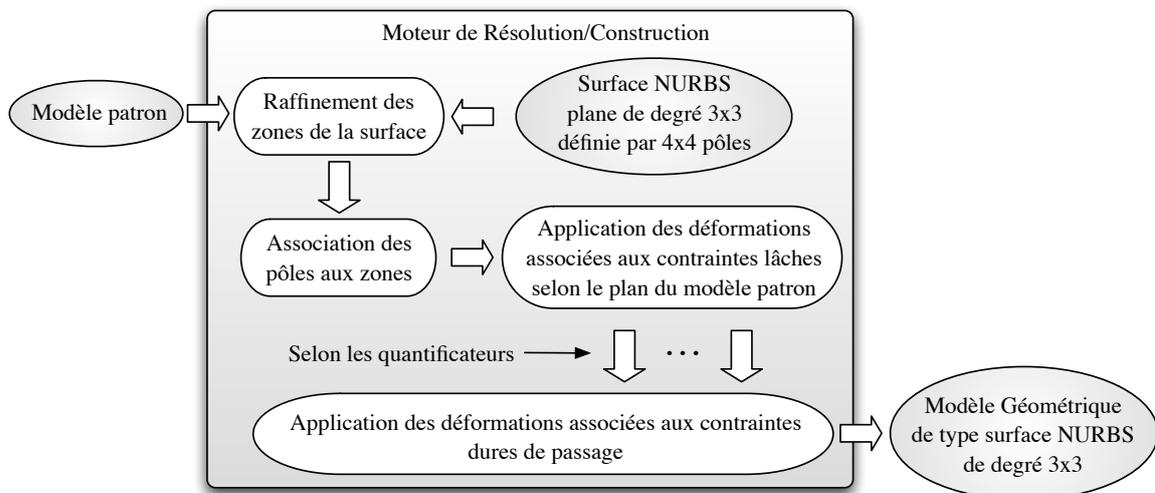


Figure 4.69. Principe du moteur de Résolution/Construction

Comme le montre la figure 4.69, notre moteur de résolution/construction prend en entrée un modèle patron. Celui-ci intègre à la fois les données sémantiques généralisées et la planification des déformations à effectuer sur une surface plane pour obtenir un modèle géométrique représentatif d'une classe de solutions. Comme cela a déjà été évoqué au début de la partie 4.3, le moteur commence par une instance simple de surface NURBS plane de degré 3×3 définie par un polyèdre de contrôle constitué de 4×4 pôles. Cette surface est en premier lieu adaptée à la taille et à la forme de la zone racine de la surface. Elle voit ensuite son polyèdre de contrôle s'enrichir de pôles par le procédé de raffinement présenté au paragraphe 4.3.3. Ce raffinement s'effectue au niveau des zones devant être déformées, en fonction du type de déformation à appliquer ou de la spécification précise de l'utilisateur. Ces nouveaux points de contrôle sont ensuite associés à chaque zone de la surface. Cette association passe par la création, pour chaque zone, d'une carte de distance, dans l'espace paramétrique, de ses frontières aux images paramétriques des pôles qui lui sont rattachés. La surface est alors prête à être déformée. On lui applique donc, selon le plan défini par le modèle patron, l'ensemble des déformations qui figurent dans sa description sémantique à l'aide des procédés définis dans la partie 4.3.4 de ce document. L'évaluation des quantificateurs (Cf. 4.3.2) lors de chaque déformation fait qu'il est possible d'obtenir des modèles géométriques différents à chaque lancement du moteur de résolution/construction puisque celui-ci génère une valeur numérique à partir d'une donnée initialement floue. Dans le cas où aucune information n'est donnée comme floue dans la sémantique du modèle (écart-type = 0 pour les quantificateurs), le moteur de résolution/construction générera toujours la même instance du modèle patron considéré.

À ce stade, la surface représente donc déjà une ébauche de la forme que souhaite l'utilisateur. Si celui-ci n'a pas spécifié de contraintes dures, comme le passage par un point particulier de l'espace par exemple, c'est ce modèle géométrique qui lui est présenté comme représentant du modèle patron considéré. Dans le cas de la précision de contraintes dites *dures*, la surface courante fait l'objet de l'une des méthodes de résolution de contraintes présentées dans la partie 4.3.6.2. Nous utilisons actuellement la méthode séquentielle qui est plus rapide que la méthode globale, et choisissons d'appliquer la résolution des contraintes de passage après l'ensemble des déformations, puisque ce sont elles qui doivent être parfaitement satisfaites en priorité.

4.4 Conclusion

Le processus de construction des modèles géométriques est l'étape finale du processus de modélisation déclarative de surface avant la proposition des solutions à l'utilisateur. Dans notre approche, cette étape est gérée par le moteur de résolution/construction qui prend en entrée un modèle patron et qui génère un modèle géométrique de type surface NURBS de degré 3×3 . Ce modèle géométrique est une instance possible de la classe de solutions que

représente le modèle patron considéré. L'utilisation de quantificateurs flous dans la description permet d'obtenir pour un même modèle patron plusieurs variantes à chaque lancement du moteur de résolution/construction. Nous avons détaillé dans ce chapitre, chaque étape du moteur que nous utilisons. Ainsi plusieurs techniques de déformation ont été proposées pour construire la surface et ainsi satisfaire l'ensemble des contraintes *lâches* attendues par l'utilisateur. Deux méthodes itératives ont aussi été présentées afin de résoudre les contraintes de passages demandées par l'utilisateur tout en contrôlant l'influence des déformations apportées à la surface.

Il est à présent possible d'obtenir les premiers résultats géométriques. C'est pourquoi, nous terminons enfin ce document par l'illustration de l'ensemble du processus déclaratif de surfaces proposé à travers quelques exemples concrets qui illustrent en même temps la maquette de modéleur déclaratif qui a été développée.

RÉSULTATS & APPLICATIONS

5.1 Introduction

Nous proposons un processus déclaratif dont l'objectif est de modéliser des surfaces paramétriques. Ce système constitue un outil d'aide à la conception permettant de réaliser des surfaces par la donnée d'une description mettant en avant des propriétés et caractéristiques particulières de la forme désirée.

Cette étude a donné lieu au développement de deux applications mettant en œuvre notre démarche conceptuelle. La première partie de ce chapitre est destinée à rapidement présenter ces deux applications. La première est une maquette de modèleur déclaratif qui prend en entrée un modèle sémantique et génère plusieurs modèles géométriques de surface paramétrique. La seconde met en application les méthodes de déformation par contraintes ponctuelles de passage que nous avons développées dans le chapitre précédent. Elle permet ainsi de *sculpter* des surfaces paramétriques par la satisfaction de différentes contraintes de passage en contrôlant leur influence.

La deuxième partie de ce chapitre présente des résultats qu'il est possible d'obtenir avec la maquette actuelle de notre modèleur. Deux exemples pratiques sont mis en évidence. Nous commençons par reproduire une surface proche de celle d'un capot de voiture puis nous nous attelons à une partie du carénage d'une moto.

5.2 Les applications développées

Au cours de cette thèse, nous avons développé deux applications : *Declarative NURBS designer* et *NURBS carver*, mettant en pratique nos études conceptuelles. Ces deux applications utilisent deux bibliothèques que nous avons développées en C++ nommées *R3D*¹ et *RDS*². *R3D* est une bibliothèque graphique de haut niveau qui utilise les bibliothèques graphiques *OpenGL*³ [Woo and al., 1999], *GLUT*⁴ et *GLUI*⁵. La bibliothèque *RDS* est une

¹Ralph third Dimension

²Ralph Declarative Surface

³site officiel : www.opengl.org

⁴OpenGL Utility Toolkit

⁵GLUT-Based User Interface Library

sur-couche de la bibliothèque *R3D* qui implémente une grande partie du processus déclaratif que nous proposons. La figure 5.1 présente les couches d'utilisation de ces bibliothèques.

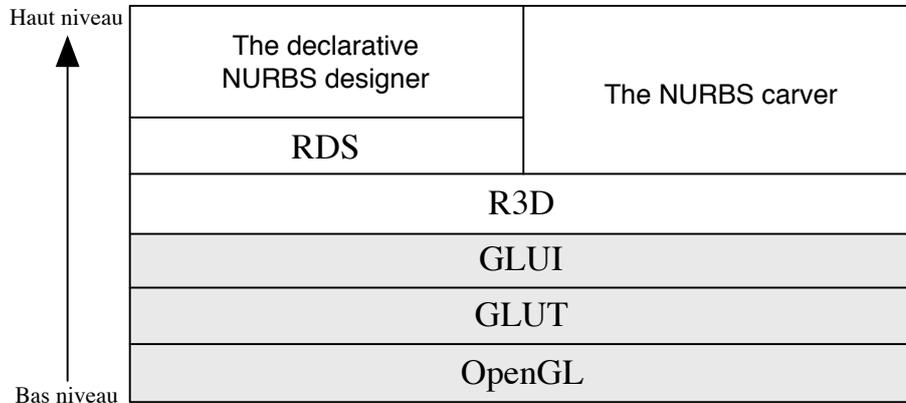


Figure 5.1. Applications et bibliothèques utilisées

L'ensemble des applications et bibliothèques développées dans le cadre de cette thèse sont disponibles sous les versions 32-bits de MS Windows[®], Linux et MacOS X.

Nous introduisons en quelques lignes ces implémentations dans les paragraphes suivants. Les présentations ne sont pas des documentations des applications considérées, mais permettent simplement de donner une idée des réalisations pratiques de notre étude. Nous ne rentrons donc pas dans les détails du code et de l'architecture qui ont permis de les obtenir.

5.2.1 La bibliothèque *R3D*

La bibliothèque *R3D* est une plate-forme de développement permettant de réaliser rapidement des applications graphiques 3D. Développée comme une sur-couche de la bibliothèque *GLUI*, elle permet de gérer les évènements système ainsi que l'interface graphique globale d'une application. Conçue dans l'optique de supporter des applications 3D, elle joue le rôle de *boîte à outils* dont les orientations sont la modélisation géométrique et l'imagerie numérique. Elle propose un ensemble d'outils permettant le traitement des images, la manipulation de maillages triangulaires et la manipulation de courbes et surfaces NURBS.

C'est dans cette bibliothèque que sont implémentées toutes les heuristiques de déformation décrites dans la partie 4.3.4, ainsi que les méthodes de satisfaction de contraintes ponctuelles détaillées à partir du paragraphe 4.3.5.

5.2.2 La bibliothèque *RDS*

La bibliothèque *RDS* s'appuie sur la bibliothèque *R3D*. Elle adapte les modèles de surfaces paramétriques à l'approche déclarative que nous proposons. *RDS* décrit ainsi une

surface comme un ensemble de zones attachées à des déformations et à des contraintes. Cette bibliothèque n'intègre à ce jour qu'une partie de la complétion sémantique. Elle ne traite en effet que les complétions, la symbolique et la numérique, des formes de la zone racine de la surface. Une perspective prioritaire de développement est d'appliquer l'ensemble des règles proposées dans les parties 3.2.4.1 et 3.2.4.2 à toutes les zones d'un modèle sémantique afin de gérer la totalité du processus de complétion sémantique.

RDS intègre de façon complète le moteur de planification. Il est ainsi possible de générer un ensemble de modèles patrons à partir d'un modèle sémantique généralisé.

Enfin, *RDS* gère la base de connaissances, les modèles sémantiques généralisés ainsi que tout le processus de résolution/construction détaillé dans la partie 4.3.8. C'est donc à partir de *R3D* et de *RDS* que la maquette de modeleur déclaratif a été conçue. Celle-ci est présentée dans le paragraphe suivant.

5.2.3 La maquette de modeleur déclaratif : «*Declarative NURBS designer*»

N'étant qu'une maquette, *Declarative NURBS designer* n'a pas vocation à intégrer toutes les fonctionnalités d'un modeleur déclaratif. Cette application est développée à partir des bibliothèques *RDS* et *R3D*. Afin de suivre la philosophie de l'approche déclarative, son interface est simple et ne demande à l'utilisateur qu'un modèle sémantique en entrée. Le processus de complétion sémantique n'étant actuellement pas complètement déployé dans *RDS*, l'application ne peut aboutir à des résultats que si on lui propose un modèle sémantique généralisé en entrée. *Declarative NURBS designer* se compose de deux fenêtres : la fenêtre de *commandes* présentée sur la figure 5.2 et la fenêtre de *rendu* illustrée sur la figure 5.3.

La fenêtre de *commandes* se divise en trois parties :

1. *Fichier* : elle permet à l'utilisateur de donner son modèle sémantique sous la forme d'un fichier XML ;
2. *Affichage* : elle donne à l'utilisateur les moyens de pouvoir observer la surface solution courante. Il lui est ainsi possible de prendre en compte les restrictions, d'afficher la surface en mode filaire, d'afficher le polyèdre de contrôle avec ses pôles et leur normale (Cf. 4.3.4.1). Mais il est aussi possible de visualiser les zones reportées sur la surface avec leur polygone de contrôle ou leur boîte englobante. L'ensemble de ces outils permet à l'utilisateur de bien prendre connaissance de la ou des solutions proposées ;
3. *Solutions* : elle présente à l'utilisateur les modèles sémantiques généralisés obtenus après la phase de complétion sémantique. Et à chacun de ces modèles, est associé un certain nombre de modèles patrons. C'est une instance du modèle patron considéré dans cette partie qui est affiché dans la fenêtre de rendu.

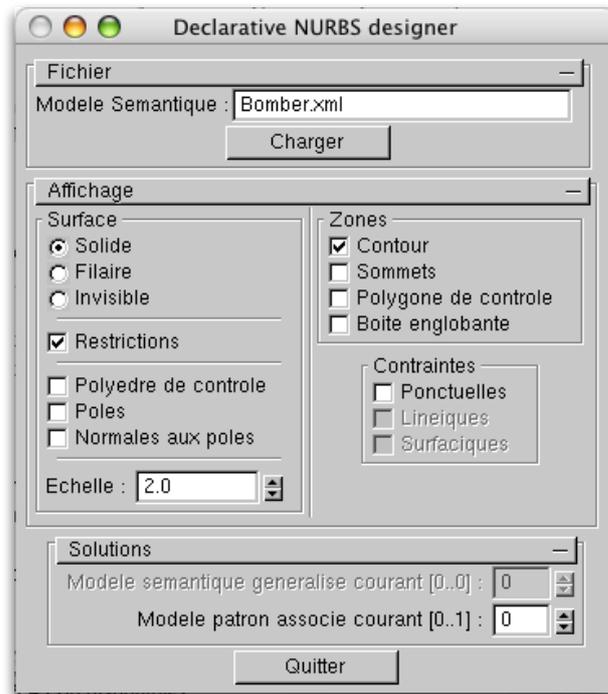


Figure 5.2. *Declarative NURBS designer* : fenêtre de commandes

La fenêtre de *rendu* est la partie qui permet de visualiser la surface solution, instance du modèle patron sélectionné. L'utilisateur peut agir avec la souris sur cette fenêtre afin d'orienter la surface selon l'angle de vue qu'il considère être le plus pertinent.

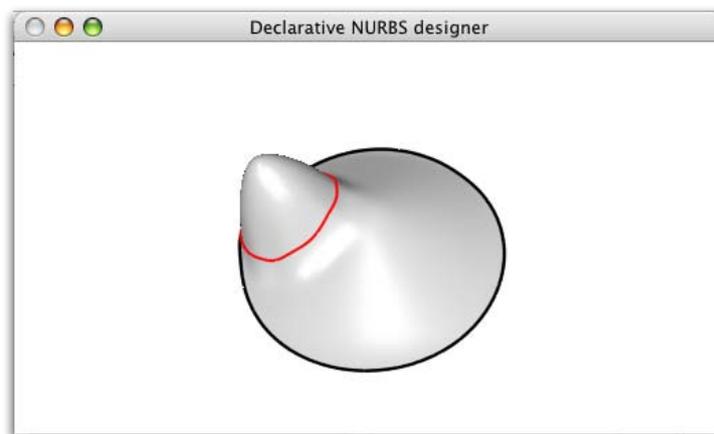


Figure 5.3. *Declarative NURBS designer* : fenêtre de rendu

Afin de suivre les opérations réalisées par l'application, il est possible de faire apparaître la *console*. Celle-ci garde une trace de tout ce que fait le processus déclaratif afin que l'utilisateur confirmé puisse en savoir plus sur le modèle résultat. D'un point de vue pratique, elle permet aussi de personnaliser l'application en «dialoguant» avec le système et en lançant

directement des commandes sans passer par l'interface de la fenêtre de commande. Elle n'est pas indispensable à l'utilisation *classique* de l'application, mais elle pourrait servir dans l'évolution de notre processus à agir sur la description au travers d'un langage pseudo naturel par exemple. La figure 5.4 illustre la console en présentant les opérations réalisées pour obtenir le modèle géométrique présenté.

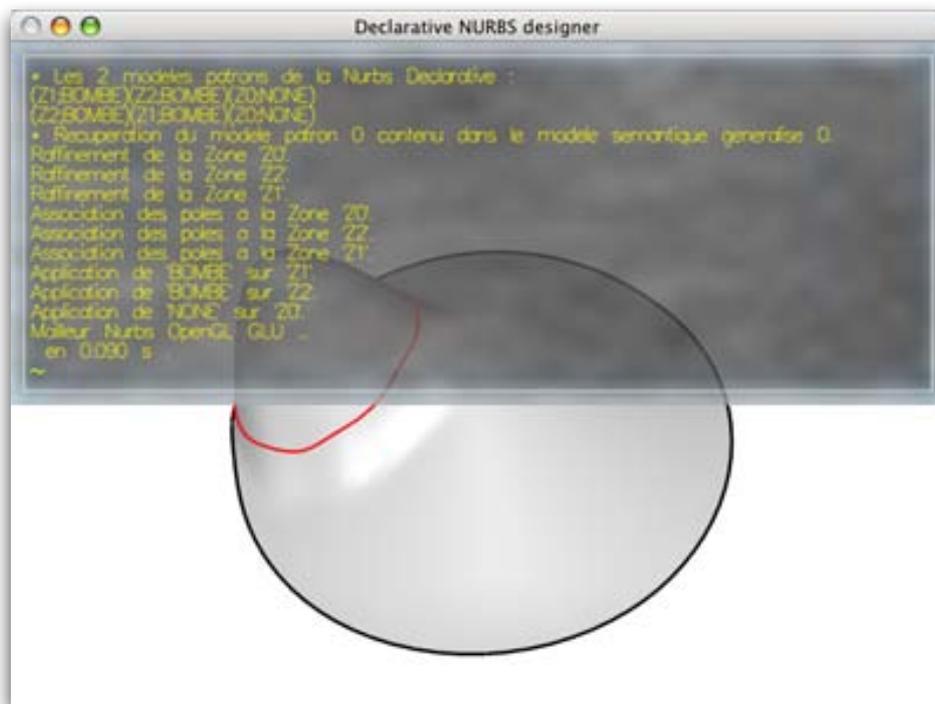


Figure 5.4. *Declarative NURBS designer* : console des opérations effectuées

Declarative NURBS designer est une application qui est amenée à évoluer, d'une part à travers les avancées des bibliothèques sur lesquelles elle repose et d'autre part afin de proposer à l'utilisateur des fonctionnalités toujours plus nombreuses. Ainsi, dans des versions futures, il sera possible d'enregistrer les modèles sémantiques et géométriques résultats que le concepteur aura choisis.

5.2.4 Le sculpteur de surfaces : «*NURBS carver*»

Le sculpteur de surface ou *NURBS carver* est une application qui a été originellement conçue pour tester et mettre au point nos méthodes de satisfaction de contraintes ponctuelles sur les surfaces NURBS. Au fur et à mesure de ses évolutions, elle s'est révélée être un outil pratique permettant de modéliser relativement rapidement une surface en considérant une ou plusieurs contraintes comme des *outils*. Cette application est encore jeune et demande à beaucoup évoluer en terme d'interface avant d'aboutir à un produit intuitif pour l'utilisateur. De la même manière que *Declarative NURBS designer*, *NURBS carver* est

constituée de deux fenêtres, l'une de *commandes* et l'autre de *rendu* comme on peut le constater sur le figure 5.5.

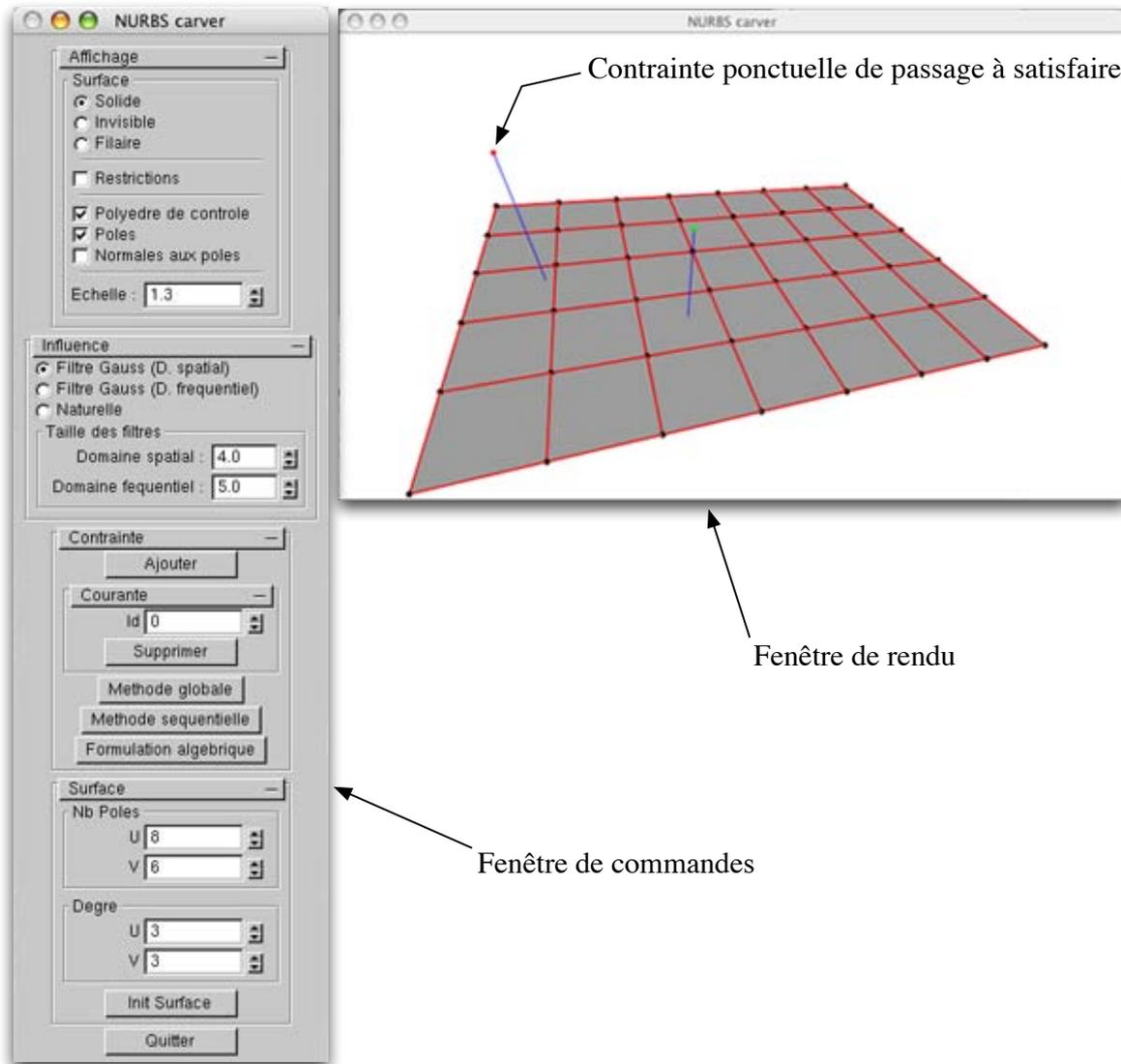


Figure 5.5. *NURBS carver*

La fenêtre de *commandes* est riche et, contrairement à l'application précédente, donne à l'utilisateur la possibilité d'agir directement sur la surface. Elle se divise en quatre parties :

1. *Affichage* : Elle permet à l'utilisateur de mieux visualiser la surface à travers plusieurs options disponibles comme l'affichage du polyèdre de contrôle et des pôles ;
2. *Influence* : C'est dans cette partie que l'utilisateur précise l'influence des contraintes. Dans une problématique de sculpture, l'influence pourrait être directement attachée à la notion d'outil et à la forme de l'impact de cet outil sur une surface. Dans la

version actuelle de l'application, l'interface ne permet pas de préciser une influence spécifique par contrainte. De plus, une seule influence est disponible. Elle consiste à élargir l'influence *naturelle* des contraintes en utilisant les procédés de filtrage par convolution cités dans la partie 4.3.5.1.2. Il faudrait donc permettre à l'utilisateur de pouvoir complètement personnaliser une contrainte ou lui permettre de choisir parmi plusieurs déjà prédéfinies. Des perspectives d'évolution de l'interface vont dans ce sens pour les prochaines versions de *NURBS carver*;

3. *Contrainte* : Il est ici possible à l'utilisateur de gérer les contraintes. Il peut en ajouter et en supprimer, mais il peut aussi lancer le processus de résolution qu'il désire en choisissant entre la méthode globale, la méthode séquentielle ou la formulation algébrique. Le résultat de l'opération de la figure 5.5 après exécution de la méthode globale correspond à la figure 5.6. C'est aussi dans cette partie que l'utilisateur pourra, dans les versions futures, donner directement les coordonnées exactes de la contrainte courante dans l'espace cartésien. Mais il pourra aussi, s'il le désire, préciser les contraintes paramétriques ainsi que la précision minimale qu'il désire localement obtenir ;

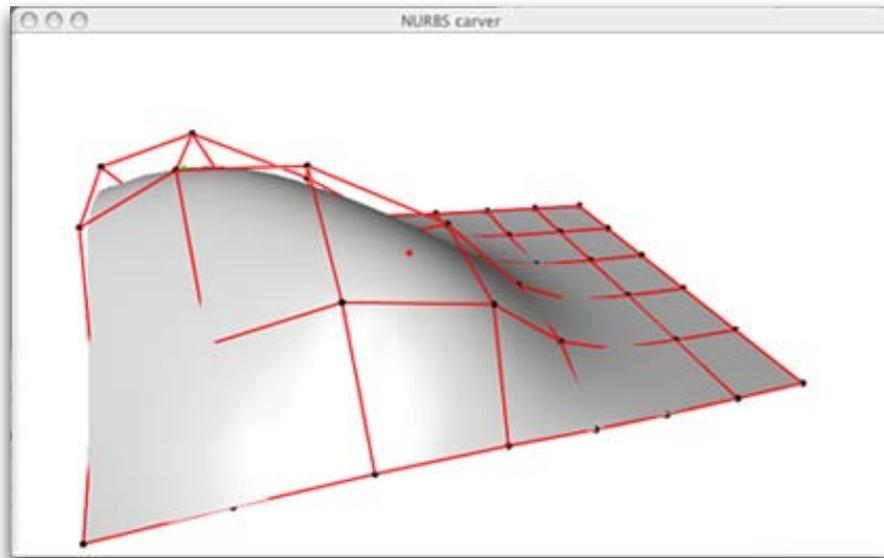


Figure 5.6. Résultat après satisfaction des contraintes avec *NURBS carver*

4. *Surface* : Par cette partie de l'interface, l'utilisateur peut directement choisir la grille de contrôle de départ ainsi que les degrés de la surface. Remarquons cependant que s'il précise l'une des directions de la grille comme étant égale à 1, il travaille directement sur une courbe gauche comme cela est illustré sur la figure 5.7.

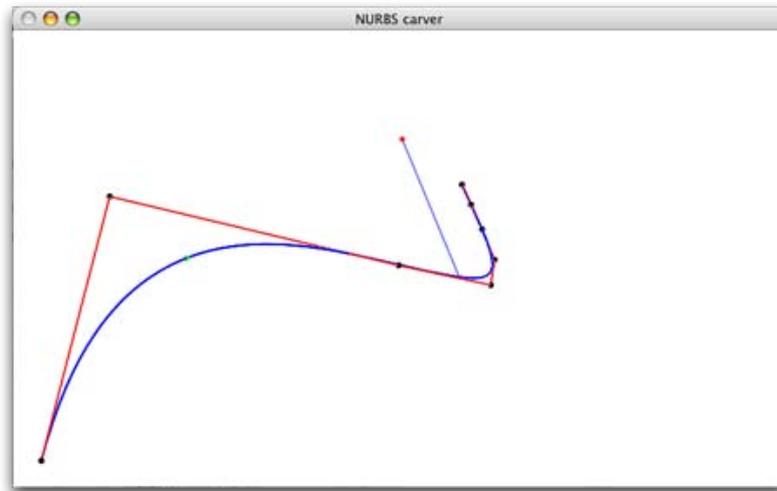


Figure 5.7. Manipulation de courbes gauches avec *NURBS carver*

En plus d'afficher la surface à déformer et de pouvoir l'orienter par l'intermédiaire de la souris, la fenêtre de *rendu* permet de positionner interactivement dans l'espace, les contraintes à satisfaire. Dans la version actuelle de l'application, les contraintes ponctuelles sont placées par l'intermédiaire des flèches directionnelles du pavé numérique permettant de se déplacer dans les trois dimensions. Un positionnement utilisant une interface homme-machine plus facile à manipuler est une perspective de développement.

NURBS carver est encore une application naissante qui va évoluer avec les avancées des bibliothèques dont elle dépend. De nombreuses perspectives de développement tournent autour de son interface homme-machine afin de donner à l'utilisateur un véritable outil interactif de manipulation de courbes et surfaces NURBS.

Les bibliothèques que nous avons développées, mises en œuvre par l'intermédiaire de l'application *Declarative NURBS designer*, ont permis d'obtenir très rapidement des résultats intéressants. Nous présentons dans la partie suivante deux exemples de surfaces paramétriques obtenues grâce à notre maquette de modelleur déclaratif de surfaces NURBS.

5.3 Quelques résultats

Cette partie présente quelques résultats obtenus en utilisant les versions actuelles de *R3D*, *RDS* et *Declarative NURBS designer*. Ne disposant pas encore, dans la version actuelle de *RDS*, de module d'interfaçage permettant d'interpréter un texte en langage naturel ou pseudo naturel, nous donnons directement à l'application *Declarative NURBS designer* un modèle sémantique généralisé. Celle-ci nous présente alors plusieurs modèles patrons susceptibles de répondre à notre demande. Un modèle géométrique, correspondant à une instance de chaque modèle patron est ainsi affiché dans la fenêtre de rendu de notre application.

Le premier exemple proposé consiste à modéliser une surface proche de celle d'un capot de Twingo[®], exemple déjà étudié lors de notre travail de description dans la partie 2.5.5. Le deuxième exemple, quant à lui, a pour objectif d'obtenir une surface se rapprochant de celle d'une partie du carénage d'une moto proposé en modèle.

5.3.1 Exemple 1 : Capot de Twingo[®]

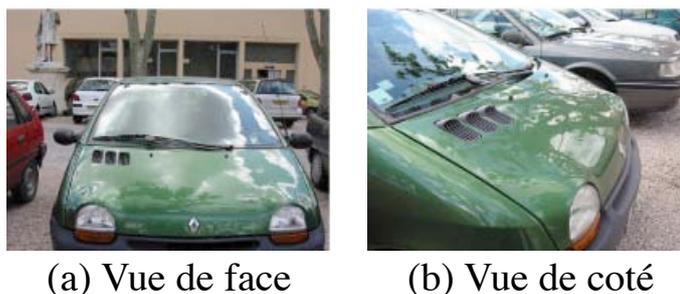


Figure 5.8. Forme de la surface que l'on souhaite obtenir : un Capot de Twingo[®]

La figure 5.8 rappelle la forme de la surface dont on souhaite reproduire l'ébauche de façon déclarative. Nous sommes dans l'espace cartésien et l'application oriente par défaut la surface initiale dans le plan (x, y) avec l'utilisateur observant cette surface à partir de l'axe des z positifs. En s'inspirant de la description déjà faite de cette surface en annexe B.2.1.5, nous proposons deux nouvelles descriptions possibles. Nous commençons par celle-ci :

- « – La surface $Z0$ est très peu large ;
- Le bord supérieur de $Z0$ est peu arrondi vers l'intérieur ;
- Le bord inférieur de $Z0$ a deux encoches $E1$ et $E2$ en forme d'arc de cercle ;
- $E1$ est sur la partie très à gauche du bord inférieur de $Z0$;
- $E2$ est sur la partie très à droite du bord inférieur de $Z0$;
- La très grande partie basse de $Z0$, notée $Z1$, est très peu courbée vers le bas ;
- $Z1$ a dans son angle carrée Haut-Gauche trois trous débouchants indépendants notés $Z6$, $Z7$ et $Z8$;
- $Z6$, $Z7$ et $Z8$ sont assez allongés vers le haut, leurs frontières inférieures et supérieures sont arrondies ;
- $Z6$, $Z7$ et $Z8$ sont parallèles suivant la longueur de $Z1$;
- $Z6$ est légèrement plus haut que $Z7$;
- $Z7$ est légèrement plus haut que $Z8$;
- $Z7$ est entre $Z6$ et $Z8$;
- La bande horizontale très inférieure de $Z1$, notée $Z2$, est courbée ;
- $Z2$ contient en son milieu une forme $Z5$ bombée ;
- $Z5$ est en forme de losange peu allongé vers le haut et troué en son centre par un losange plus petit ;

- Une partie carrée de la zone Bas-Gauche de Z1 qui coïncide avec E1 est extrêmement peu bombée ;
- Une partie carrée de la zone Bas-Droite de Z1 qui coïncide avec E2 est extrêmement peu bombée ;

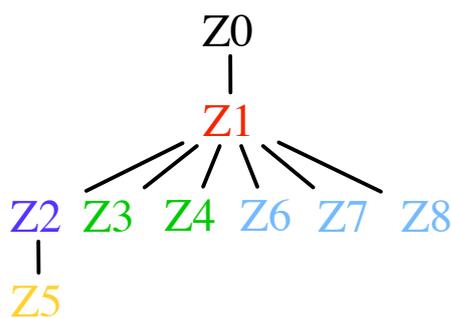
»

Cette première description peut se représenter selon l'arbre structurant \mathcal{A}_s de la figure 5.9.a. Cependant, si les deux dernières lignes sont changées en :

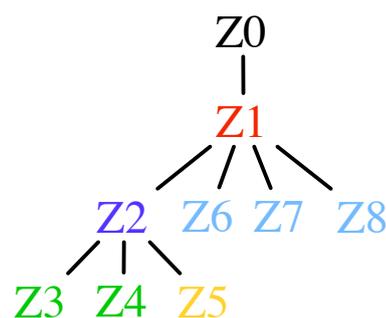
- « - Une partie carrée de la **bande verticale gauche de Z2** qui coïncide avec E1 est extrêmement peu bombée ;
- Une partie carrée de la **bande verticale droite de Z2** qui coïncide avec E2 est extrêmement peu bombée ;

»

On est alors dans le cas d'une deuxième description qui se représente cette fois-ci par l'arbre structurant de la figure 5.9.b.



(a) As de la description 1



(b) As de la description 2

Figure 5.9. Arbres structurants correspondants aux deux descriptions

Nous traitons ici le cas des deux descriptions. L'annexe C.3 propose un modèle sémantique généralisé qu'il est possible d'obtenir après la phase de complétion sémantique pour le cas de la description 1. La différence entre les deux descriptions réside simplement dans les relations hiérarchiques entre les zones. Dans les deux cas, la position des zones reste la même et peut se représenter comme sur la figure 5.10. Il est important de remarquer que nous gérons ici la forme arrondie du bord supérieur de Z0 comme une restriction. Cet arrondi pourrait très bien être obtenu en déplaçant les pôles définissant le bord de la surface. Cependant, les déformations attachées aux bords de la surface ne sont encore qu'une perspective de développement. Ainsi, nous choisissons pour le moment une restriction qui nous permet d'obtenir un résultat visuellement équivalent.

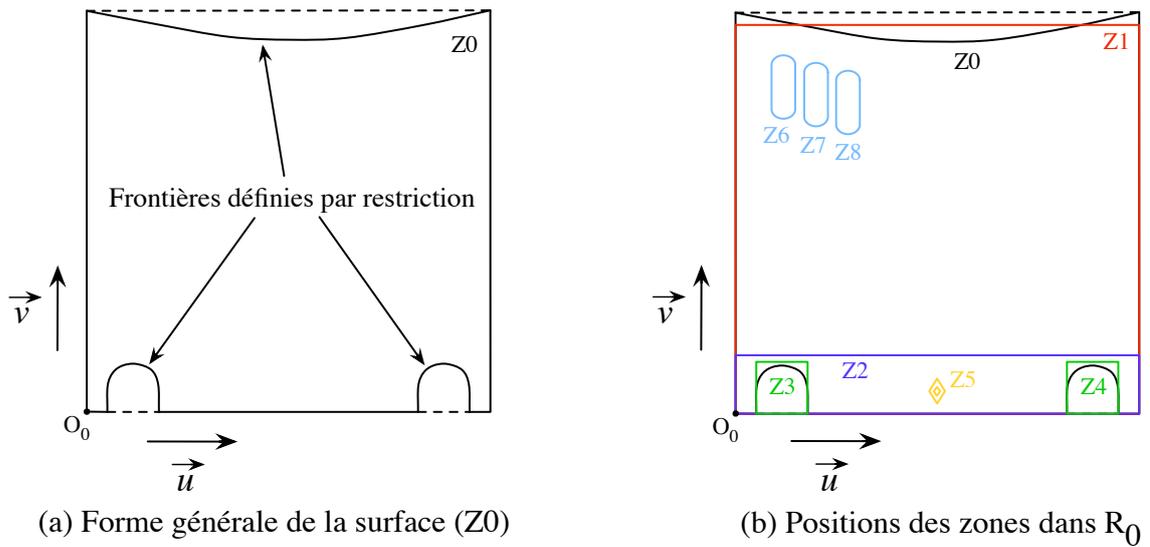


Figure 5.10. Définition des zones dans le plan paramétrique

5.3.1.1 Cas de la première description

Après chargement du modèle sémantique généralisé, *Declarative NURBS designer* trouve quatre modèles patrons, chacun associé à une des séquences suivantes :

- (Z1;Cb) (Z2;Cb) (Z6;Tr) (Z7;Tr) (Z8;Tr) (Z4;Bb) (Z3;Bb) (Z5;Bb) (Z0;No) => Patron 1
- (Z1;Cb) (Z6;Tr) (Z7;Tr) (Z8;Tr) (Z4;Bb) (Z3;Bb) (Z2;Cb) (Z5;Bb) (Z0;No) => Patron 2
- (Z1;Cb) (Z4;Bb) (Z2;Cb) (Z3;Bb) (Z5;Bb) (Z6;Tr) (Z7;Tr) (Z8;Tr) (Z0;No) => Patron 3
- (Z1;Cb) (Z3;Bb) (Z6;Tr) (Z2;Cb) (Z4;Bb) (Z5;Bb) (Z7;Tr) (Z8;Tr) (Z0;No) => Patron 4

Ces quatre séquences sont dues aux dépendances mutuelles qu'il existe entre Z2 et Z3 et entre Z2 et Z4. Chacune des figures 5.11, 5.12, 5.13 et 5.14 présente une instance possible des modèles patrons proposés, vue sans et avec le report des zones sur la surface respectivement en (a) et (b).

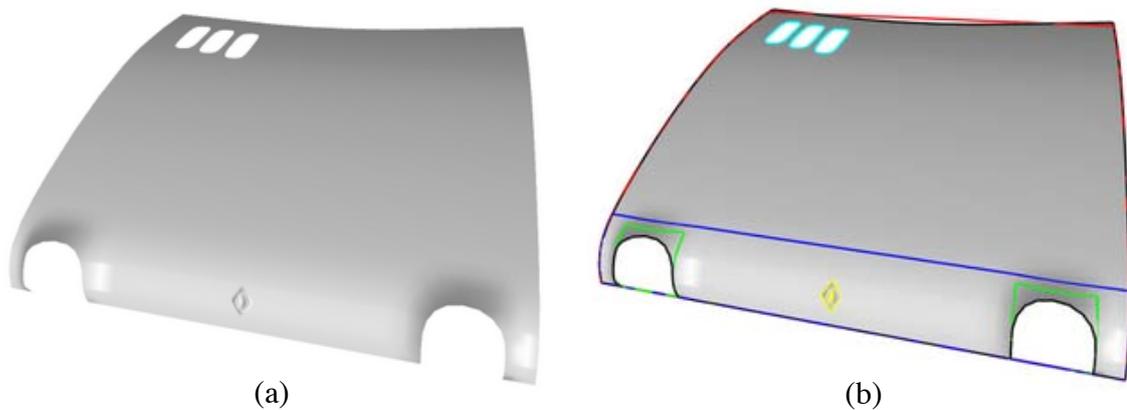


Figure 5.11. Instance possible du modèle patron 1

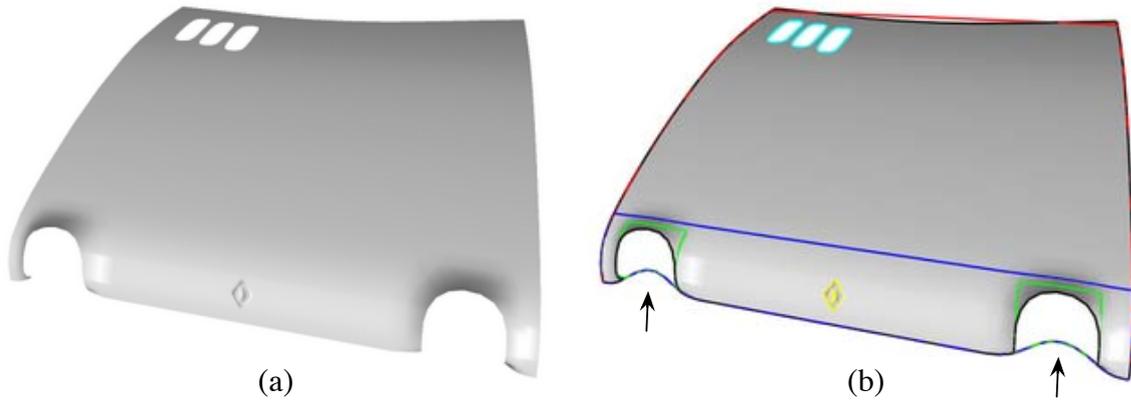


Figure 5.12. Instance possible du modèle patron 2

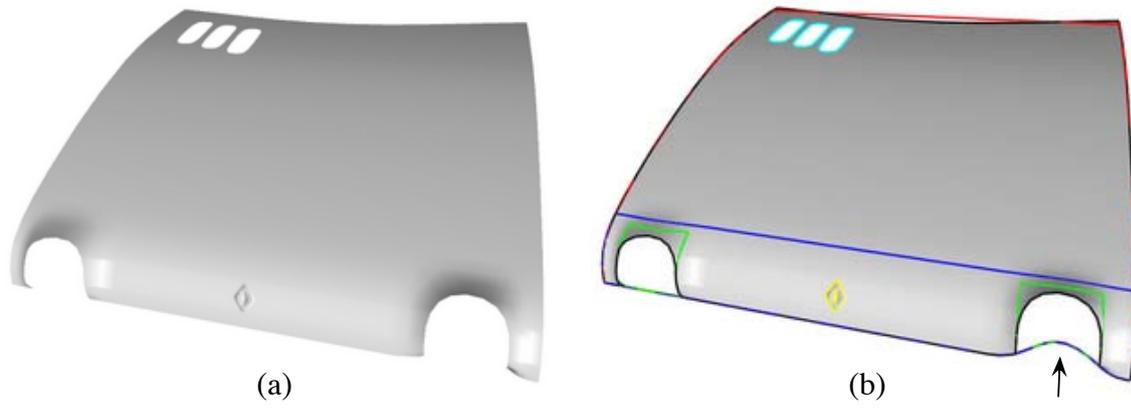


Figure 5.13. Instance possible du modèle patron 3

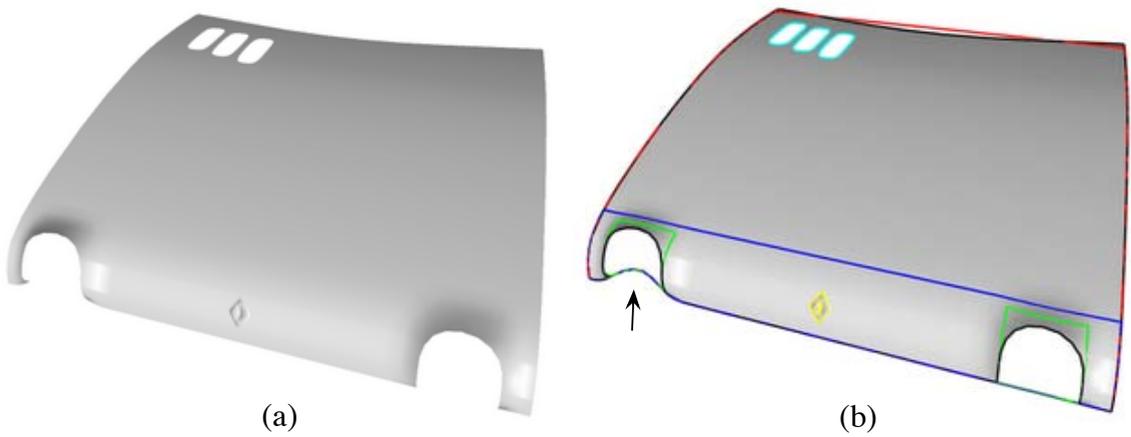


Figure 5.14. Instance possible du modèle patron 4

Les différences qui existent entre ces quatre modèles, n'apparaissent qu'aux niveaux des zones $Z3$ et $Z4$ selon que le *bombé* est réalisé avant ou après le *courbé* de la zone $Z2$. Ces différences sont pointées d'une flèche d'une figure à l'autre. Nous pouvons de plus remarquer que les différences les plus importantes, sont dans les restrictions et que, de ce fait, les surfaces résultats (a) sont visuellement semblables. Voyons maintenant le résultat de la deuxième description.

5.3.1.2 Cas de la deuxième description

Après chargement du modèle sémantique généralisé, *Declarative NURBS designer* ne trouve qu'un seul modèle patron qui est associé à la séquence suivante :

(Z1;Cb) (Z6;Tr) (Z7;Tr) (Z8;Tr) (Z2;Cb) (Z5;Bb) (Z4;Bb) (Z3;Bb) (Z0;No) => Patron 1

Ce résultat était tout à fait prévisible puisque les dépendances mutuelles existantes dans la description précédente ont été supprimées par les relations hiérarchiques puisque $Z2$ est maintenant mère de $Z3$ et $Z4$. La figure 5.15 illustre ce résultat qu'elle présente avec (a) et sans (b) le report des zones sur la surface.

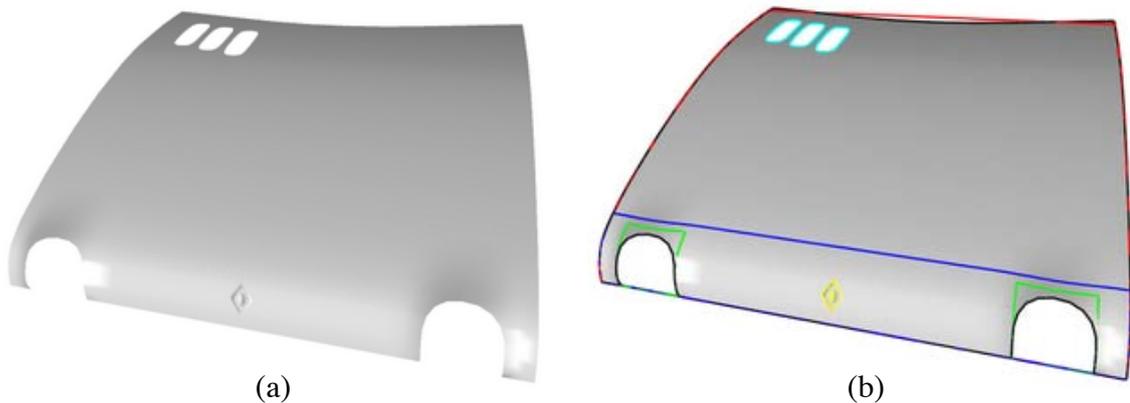


Figure 5.15. Instance possible correspondant à la deuxième description

Dans les deux cas de description, les modèles géométriques proposés par notre application sont très proches de la surface souhaitée.

5.3.2 Exemple 2 : Carénage de moto

La figure 5.16 présente le modèle de la surface que nous souhaitons obtenir. La surface est relativement complexe faisant apparaître plusieurs zones dont la forme ne correspond pas à une primitive géométrique simple⁶. La bibliothèque *RDS* étant encore à ce jour en phase de développement, elle ne permet pas de placer les pôles définissant les polygones de contrôle d'une zone dans le plan paramétrique (complétion numérique d'une zone). C'est

⁶Ex. : un triangle, un carré, un rectangle, un cercle ...

pourquoi, nous donnons directement à notre application un modèle sémantique généralisé contenant déjà leur position. Ce modèle n'est pas donné dans le document car il n'apporte pas plus d'information au lecteur que celui déjà donné en annexe pour l'exemple du capot de twingo®. Le code XML est de plus sensiblement de la même taille.

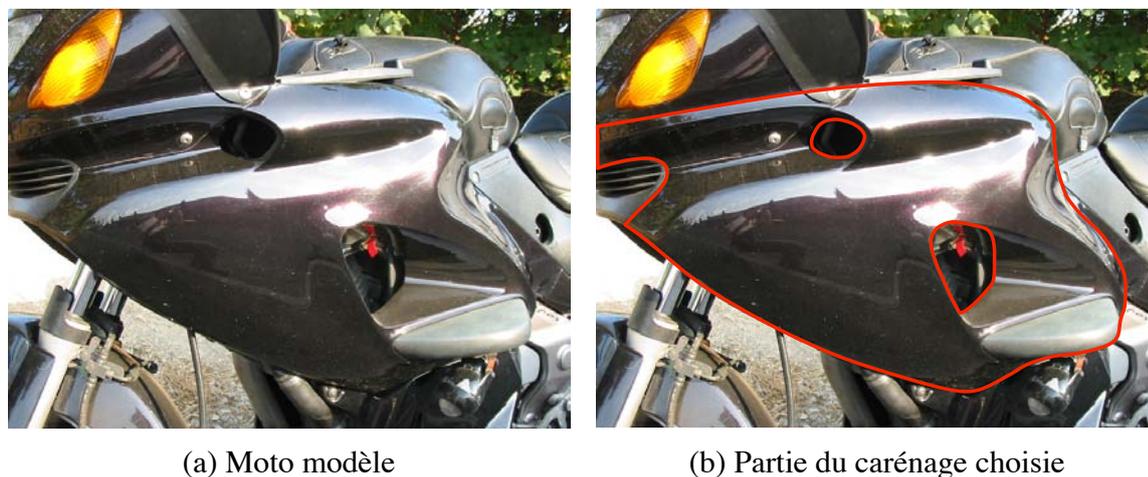


Figure 5.16. Modèle de la surface du carénage que l'on souhaite obtenir

Le figure 5.17 illustre la définition dans le plan paramétrique, des zones que l'on peut considérer sur le carénage.

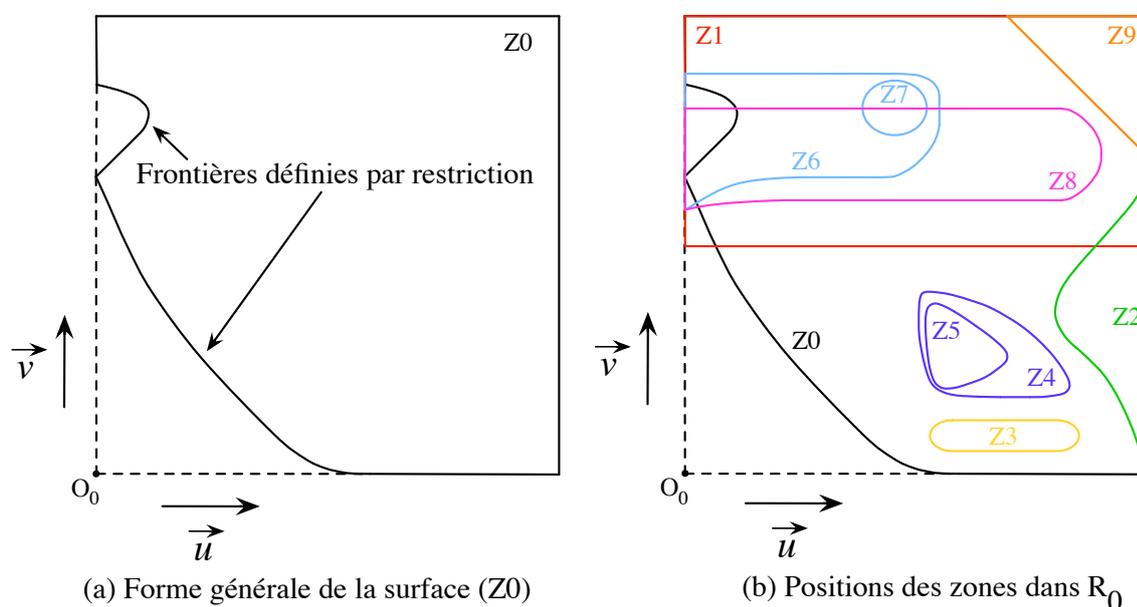


Figure 5.17. Définitions des zones du carénage dans le plan paramétrique

Ces zones respectent la hiérarchie présentée sur la figure 5.18. Nous considérons ainsi la surface comme étant courbée dans toute sa moitié supérieure. On la note $Z1$. Nous précisons que $Z1$ est arrondie dans son angle Haut-Droit triangulaire ($Z9$) et que $Z1$ contient deux

zones ($Z6$ et $Z8$) creusées très larges qui se chevauchent en partie. La zone trouée ($Z7$) à l'intérieur de $Z6$ qui est creusée, est considérée comme fille de $Z6$. De même, la zone $Z5$ trouée à l'intérieur de $Z4$ qui est creusée, est considérée comme fille de $Z4$. Enfin $Z2$ est creusée et $Z3$ est bombée.

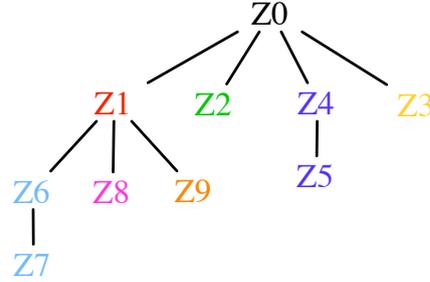


Figure 5.18. Hiérarchie des zones

En donnant à l'application *Declarative NURBS designer* toutes ces informations quantifiées sous la forme d'un modèle sémantique généralisé, celle-ci nous propose 4 modèles patrons solutions répondant aux séquences suivantes :

($Z4$;Cr) ($Z5$;Tr) ($Z3$;Bb) ($Z2$;Cr) ($Z1$;Cb) ($Z8$;Cr) ($Z6$;Cr) ($Z9$;Cb) ($Z7$;Tr) ($Z0$;No) => Patron 1
 ($Z1$;Cb) ($Z3$;Bb) ($Z8$;Cr) ($Z4$;Cr) ($Z6$;Cr) ($Z7$;Tr) ($Z2$;Cr) ($Z5$;Tr) ($Z9$;Cb) ($Z0$;No) => Patron 2
 ($Z4$;Cr) ($Z5$;Tr) ($Z3$;Bb) ($Z2$;Cr) ($Z1$;Cb) ($Z6$;Cr) ($Z9$;Cb) ($Z8$;Cr) ($Z7$;Tr) ($Z0$;No) => Patron 3
 ($Z1$;Cb) ($Z9$;Cb) ($Z6$;Cr) ($Z8$;Cr) ($Z3$;Bb) ($Z4$;Cr) ($Z2$;Cr) ($Z7$;Tr) ($Z5$;Tr) ($Z0$;No) => Patron 4

Les quatre solutions viennent du fait qu'il existe une dépendance mutuelle entre $Z1$ et $Z2$ et une autre entre $Z6$ et $Z8$. Chacune des figures 5.19, 5.20, 5.21 et 5.22 présente une instance possible des modèles patrons proposés, vue sans et avec le report des zones sur la surface respectivement en (a) et (b).

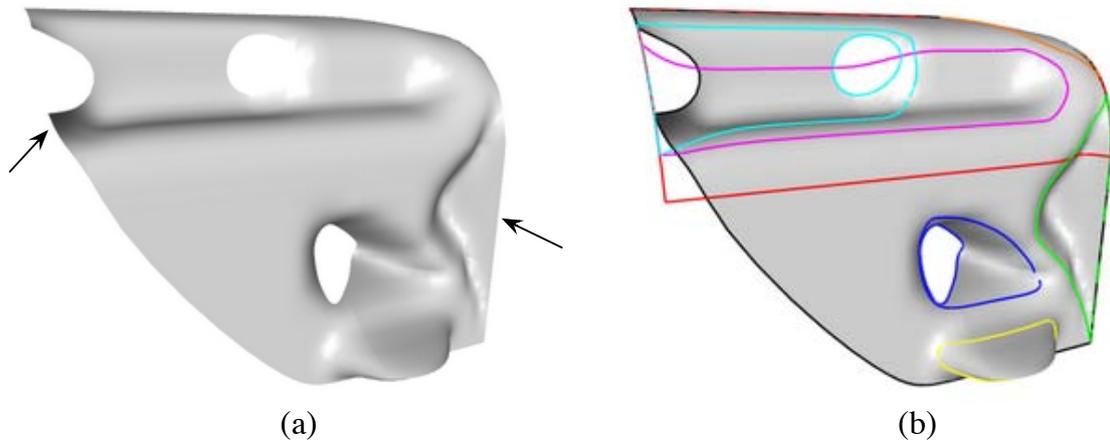


Figure 5.19. Instance possible du modèle patron 1

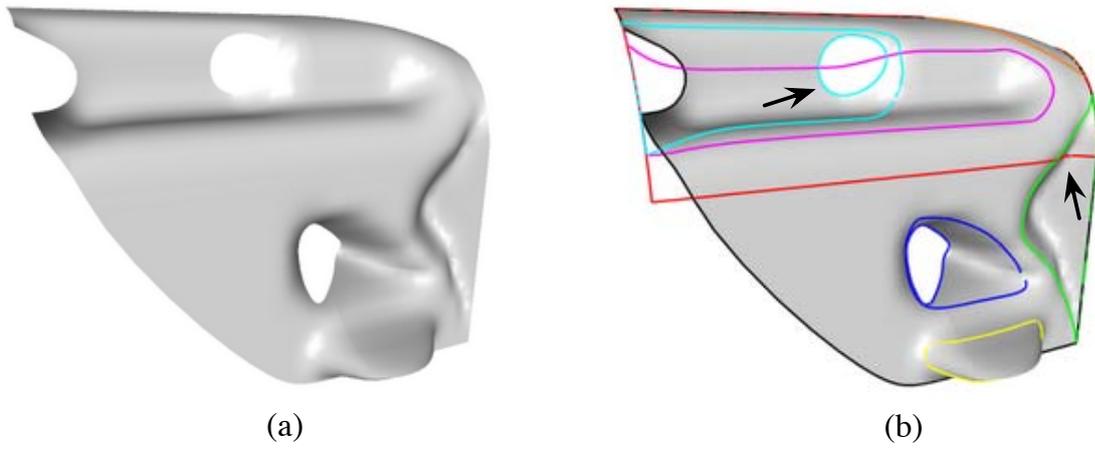


Figure 5.20. Instance possible du modèle patron 2

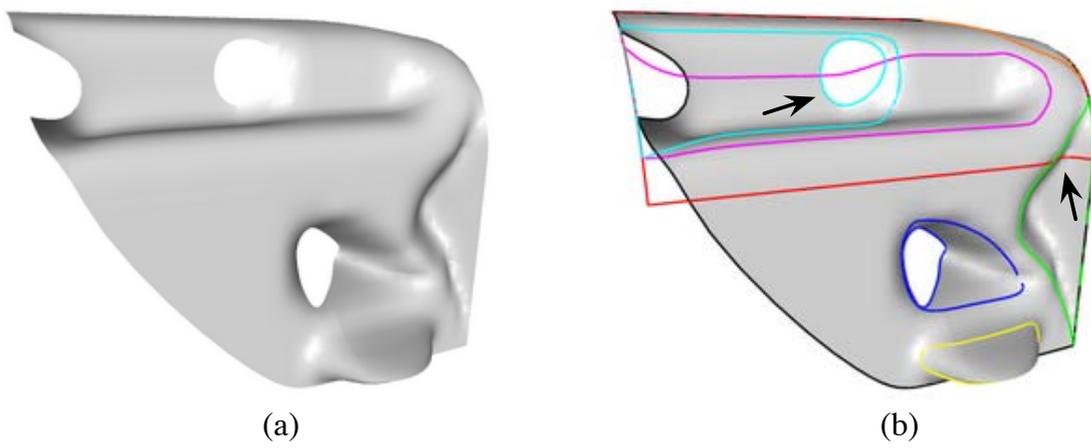


Figure 5.21. Instance possible du modèle patron 3

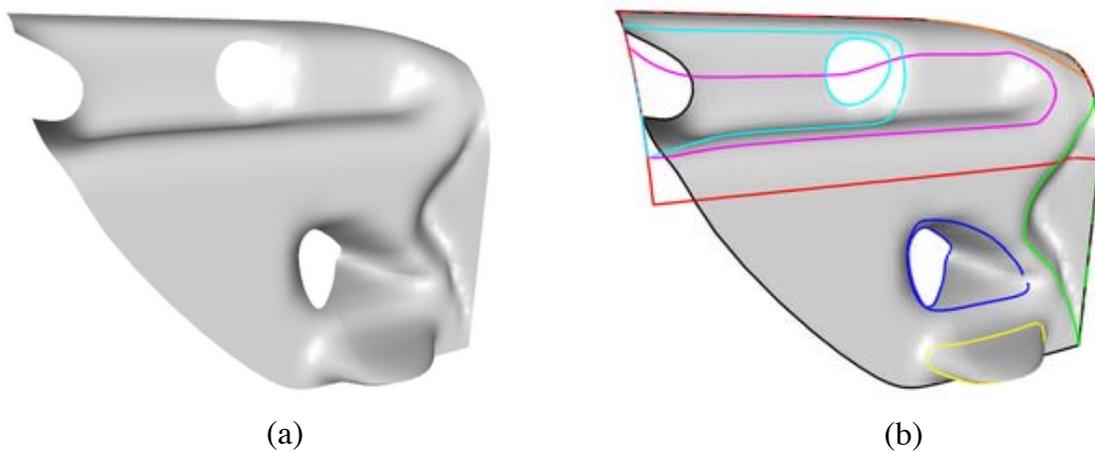


Figure 5.22. Instance possible du modèle patron 4

Nous précisons sur les figures 5.20.b et 5.21.b les endroits où les variations sont les plus fortes. Cependant, ces différences sont visuellement difficiles à repérer. En effet, la zone de recouvrement entre $Z1$ et $Z2$ est petite face à la taille des deux zones. De plus, les déformations dans cette région sont faibles. Entre $Z6$ et $Z8$, la principale influence se porte sur la forme trouée ($Z7$) à l'intérieur de $Z8$ qui apparaît plus ronde pour les modèles patrons 3 et 4 et plus large pour les modèles patrons 1 et 2.

Les résultats sont globalement proches de la surface désirée à l'exception des parties fléchées sur la figure 5.19.a. Nous remarquons une remontée aux bords de la surface alors que celle-ci n'apparaît pas sur le carénage modèle. Cela est dû au fait que la déformation «*creuser*» utilise la grille de distance paramétrique (Cf. 4.3.3) de la zone pour la déformer. Ainsi la déformation réalisée aux frontières de la zone est plus faible qu'en son centre. Il faudrait donc considérer un autre type de déformation «*creuser*» qui ne tienne pas compte des bords des zones confondus avec ceux de la surface. Cela correspond au concept de *configuration* de déformations évoqué dans la partie 4.3.4.6 qui constitue une perspective d'évolution de notre travail. Le figure 5.23 représente les quatre modèles patrons précédents en considérant ce nouveau type de «*creuser*».

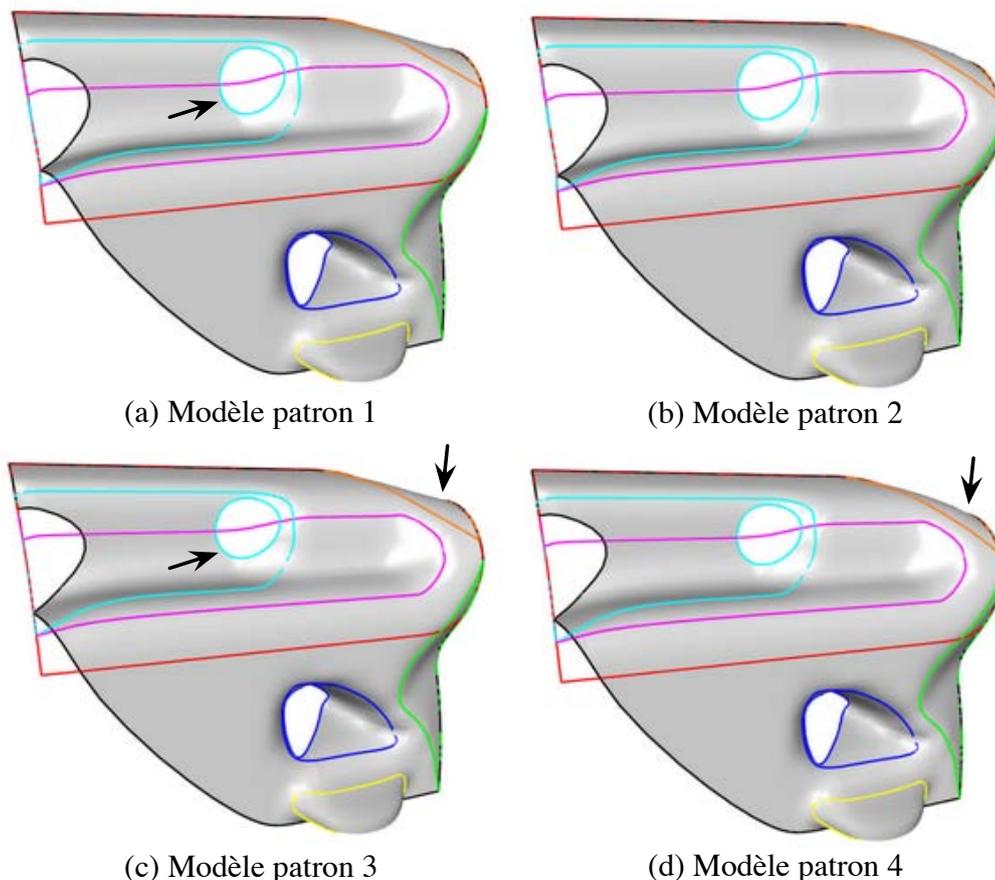


Figure 5.23. Instances des modèles patrons avec la nouvelle déformation «*creuser*»

Les résultats obtenus sont de bonnes ébauches du carénage modèle. La figure 5.24 présente le modèle patron 4 vu de l'intérieur sans et avec le report des zones sur la surface respectivement en (a) et (b).

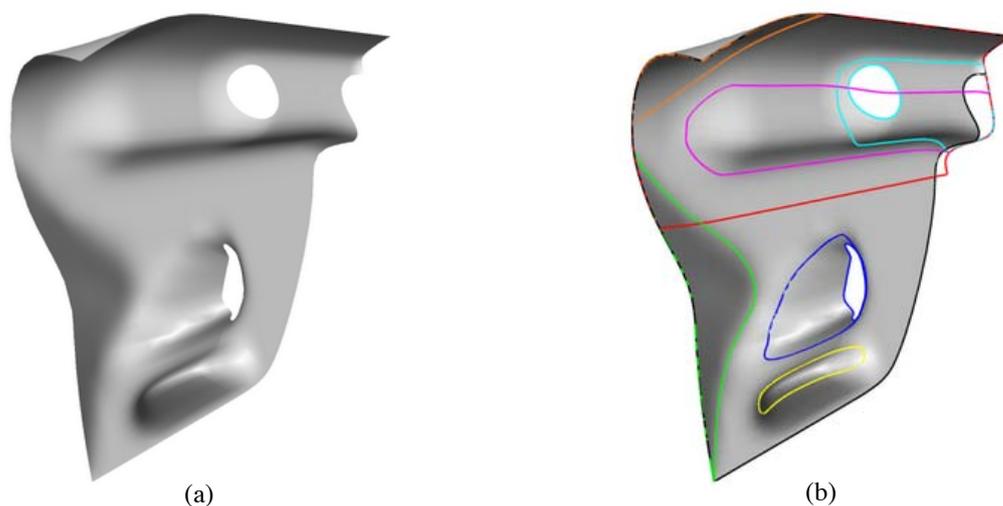


Figure 5.24. Modèle patron 4 vu de l'intérieur

5.4 Conclusion

Ce chapitre a mis en valeur les réalisations pratiques de l'étude proposée dans ce document. Ces recherches ont abouti au développement de deux bibliothèques et de deux applications. La première bibliothèque nommée *R3D* est basée sur OpenGL et permet la création rapide d'applications graphiques 3D. Elle propose entre autres la manipulation de courbes et surfaces NURBS ainsi que les méthodes de déformation par satisfaction de contraintes ponctuelles de passage développées dans le chapitre 4. La deuxième bibliothèque, nommée *RDS*, repose sur *R3D* et implémente une grande partie du processus déclaratif présenté dans ce document. Enfin, en s'appuyant sur ces deux bibliothèques, nous avons mis au point deux applications. L'une, appelée *Declarative NURBS designer*, exploite les possibilités déclaratives de nos bibliothèques. L'autre, nommée *NURBS carver*, est une application naissante et permet de «sculpter» des surfaces NURBS à partir des méthodes de satisfaction de contraintes déjà citées. Deux exemples pratiques, consistant à modéliser un capot de voiture et une partie du carénage d'une moto, ont ensuite été présentés en utilisant *Declarative NURBS designer*. Les résultats obtenus sont des ébauches de surfaces proches des surfaces modèles, ce qui est l'objectif de la modélisation déclarative.

SYNTHÈSE ET PERSPECTIVES

Synthèse

L'étude proposée dans ce document a pour objectif de mettre en place une base de travail dans le domaine de la modélisation de surfaces à travers une approche dite *déclarative*. Elle en a fixé les besoins et nous avons établi un processus complet, accompagné de son architecture, spécifiquement étudiée pour la conception de surfaces. Cette étude a abouti à une maquette de modèleur et à de nombreuses perspectives de recherche que nous détaillons dans la partie suivante.

Dans le premier chapitre nous avons introduit la problématique générale de notre travail. Nous avons présenté le modèle paramétrique de type NURBS qui a été choisi pour le développement de cette thèse. Puis, après une synthèse des recherches réalisées dans le domaine de la modélisation déclarative, nous avons présenté le processus général que nous avons établi dans le cadre précis de la conception de surfaces.

Le deuxième chapitre a présenté tout notre travail basé sur l'étude de la description de surfaces. Nous avons fait la constatation d'une influence forte du domaine d'application et avons décidé de nous orienter vers la description de pièces issues du monde de la mécanique. Cela nous a amenés à l'élaboration d'un lexique et à la conception d'un pseudo-langage issu du français écrit et s'exprimant sous la forme d'une grammaire⁷ de description. Nous sommes donc partis de l'hypothèse qu'une description est suffisante pour obtenir une ou un ensemble d'ébauches de formes.

Ces descriptions étant riches en sémantique, nous avons entièrement consacré le chapitre trois à la structure des modèles que nous utilisons. Nous avons ainsi introduit la notion de modules d'interface afin de donner une dimension multimodale à la description que peut donner l'utilisateur. Nous avons aussi exprimé le cheminement des opérations que nous proposons pour obtenir un *modèle sémantique* qui se décline ensuite en plusieurs *modèles sémantiques généralisés*, puis en plusieurs *modèles patrons* après les processus de *complétion sémantique* et de *planification*.

⁷au sens de Chomsky

Le chapitre quatre s'est ensuite penché sur l'exploitation des modèles patrons par le *moteur de résolution/construction* en charge de construire la ou les surfaces résultats par un ensemble de déformations. Nous avons ainsi introduit la notion de contraintes *lâches* qui correspondent à des tendances désirées par le concepteur et de contraintes *dures* ou *strictes* qui sont des exigences précises comme le passage de la surface par un point donné de l'espace. Pour ces dernières nous avons proposé deux méthodes itératives de satisfaction de contraintes ponctuelles de passage que nous avons aussi exprimés sous une forme algébrique commune.

Dans le chapitre cinq nous avons présenté les deux bibliothèques ainsi que les deux applications qui ont été développés tout au long de cette thèse. Celles-ci nous ont permis d'obtenir des résultats intéressants que nous avons illustrés par des exemples pratiques.

Perspectives

Ce document a fait émerger de nombreuses perspectives dans le but de concevoir un modelleur déclaratif complet de surfaces NURBS. Nous présentons ici les principaux axes que nous souhaitons explorer.

- La perspective à plus court terme et qui reste à ce jour prioritaire est l'implémentation dans la bibliothèque *RDS* de l'intégralité des règles de complétion sémantique définies dans les parties 3.2.4.1 et 3.2.4.2 de ce document. Ce travail de développement permettra de directement générer des modèles sémantiques généralisés dans lesquels seront définis les polygones de contrôle des zones paramétriques de la surface. Ainsi, il sera possible de donner directement des modèles sémantiques en entrée de l'application *Declarative NURBS designer*.
- La réalisation de modules d'interfaçage est très importante. En effet, ces derniers permettent de monter à un niveau supérieur en terme d'interface Homme-Machine puisqu'ils ont pour vocation de générer des modèles sémantiques. Il est donc essentiel de poursuivre nos recherches dans le sens d'un pseudo-langage de description comme nous l'avons proposé dans la partie 2.5.4. L'objectif à long terme est l'interprétation de descriptions écrites ou orales données en plusieurs langues.
- Comme nous l'avons déjà mentionné, dans une approche déclarative, ce n'est pas le concepteur qui s'adapte aux outils, mais bien l'outil qui s'adapte au concepteur. Pour cela, l'application doit évoluer avec son utilisation. Il faut donc prévoir à long terme un système d'apprentissage. La base de connaissances se personnalisera et permettra de stocker les modifications de quantification, le nouveau vocabulaire ainsi que certaines formes définies par l'utilisateur.
- Une perspective destinée à l'évolution du modelleur consiste à l'enrichir de nouvelles déformations, ou contraintes *lâches* applicables à des zones de la surface, mais aussi à ses bords. L'évolution des déformations déjà présentes est aussi envisagée, comme

- par exemple ajouter à la déformation «courber» la possibilité d'utiliser un axe paramétrique sous la forme d'un polygones ou d'une courbe. Cette évolution passe aussi par l'introduction du concept de *configuration* ou de *propriétés* des déformations. Il existerait alors plusieurs déclinaisons d'une même déformation permettant ainsi à l'utilisateur de choisir celle qui s'adapte au mieux à sa demande.
- En tenant compte de l'ensemble des nouveaux résultats obtenus, il serait intéressant d'explorer à nouveau les possibilités d'exploitation du logiciel DEFSURF adapté à notre méthode.
 - Dans le but d'aboutir à un modéleur déclaratif permettant de satisfaire plusieurs types de contraintes que nous qualifions de *strictes*, il nous faut gérer plusieurs notions supplémentaires détaillées dans la partie 2.4.7 du chapitre 2 dont voici les principales orientations :
 - les contraintes de tangence ;
 - les contraintes surfaciques⁸ ;
 - les contraintes de non-passage ;
 - les contraintes d'aires et de volumes ;
 - les contraintes de courbures ;
 - les contraintes d'ingénierie ;
 - les contraintes fonctionnelles ;
 - les contraintes de dimensions.
 - Un point très important que doit pouvoir gérer notre système est le lien entre les modèles sémantiques et les modèles géométriques. Chaque modèle géométrique est actuellement associé à un modèle sémantique généralisé décrivant les zones de la surface, leurs déformations . . . Or il n'existe pas, à ce jour, de moyen de contrôler les déformations géométriques pour garder la cohérence avec le modèle sémantique. Ainsi, si le polyèdre est modifié, il n'y a aucun moyen de savoir si les propriétés ou les formes initiales sont encore vérifiées. Comme cela a déjà été fait dans un plan dans [Rossignol, 2001], un axe de recherche serait de définir pour chaque pôle un volume de validité pour lequel il serait possible de certifier que toutes les contraintes restent vérifiées. Ce problème est difficile et est actuellement considéré comme un verrou scientifique par la communauté travaillant dans le domaine des contraintes géométriques.
 - L'ensemble de nos calculs tient compte du poids des pôles définissant la surface. Cependant, ceux-ci ne sont jamais modifiés et sont initialement fixés à 1. Le travail revient donc à manipuler des B-splines. Une perspective importante est de pouvoir aussi intervenir sur les poids de la surface afin de véritablement travailler sur des surfaces NURBS et ainsi exploiter toutes leurs propriétés.

⁸une zone de la surface doit pouvoir épouser exactement la forme d'une autre surface donnée

- Compte tenu de l'évolution du modeleur, celui-ci va pouvoir proposer de plus en plus de solutions possibles au concepteur. L'arbre des solutions va donc être de plus en plus grand ce qui ne va pas faciliter le choix des solutions pour le concepteur. C'est pourquoi, il est essentiel de développer des outils de parcours et d'exploration de l'arbre des solutions.
- Alors que l'objectif actuel consiste à modéliser des surfaces, il est possible d'envisager à long terme de les *habiller* et de les animer. L'intégration de propriétés temporelles et visuelles peuvent ainsi être imaginées dans le procédé déclarative avec la description des mouvements, des textures et des propriétés d'illumination et de transparence des matériaux de la surface.
- Enfin, bien que ce ne soit pas directement lié à la problématique de la modélisation déclarative, les méthodes de satisfaction de contraintes ponctuelles étudiées pour les courbes et les surfaces pourraient être étendues aux volumes. Ainsi il serait possible de les appliquer à des méthodes de déformations comme les FFD que nous avons présentées dans la partie 4.2.1.2 du chapitre 4.

Toutes ces perspectives sont bien évidemment directement rattachées au développement continu des applications *Declarative NURBS designer* et *NURBS carver* qui permettent de mettre en valeur tous les résultats théoriques obtenus.

La modélisation déclarative de surfaces est un projet ambitieux qui ouvre de nombreuses perspectives de recherche. Nous pensons que l'idée est innovante et pourrait commencer à percer dans le monde de la conception d'ici quelques années.

MODÉLISATION GÉOMÉTRIQUE

A.1 Géométrie différentielle

A.1.1 Première forme fondamentale

Soit Γ une courbe définie sur \mathcal{S} et passant par M , elle s'exprime alors sous la forme $\Gamma(u(t); v(t))$. Soit $s(t)$ son abscisse curviligne, il est alors possible de calculer $\left(\frac{ds(t)}{dt}\right)^2$ et d'en déduire la première forme fondamentale de la théorie des surfaces :

$$\left(\frac{ds}{dt}\right)^2 = \left(\frac{d\overrightarrow{OM}}{dt}\right)^2 = \left[\frac{\partial\overrightarrow{OM}}{\partial u} \frac{du}{dt} + \frac{\partial\overrightarrow{OM}}{\partial v} \frac{dv}{dt}\right]^2 = \left[\overrightarrow{M}_u \frac{du}{dt} + \overrightarrow{M}_v \frac{dv}{dt}\right]^2$$

Soient :

$$\begin{cases} E = \left(\frac{\partial\overrightarrow{OM}}{\partial u}\right)^2 = \overrightarrow{M}_u^2 \\ F = \frac{\partial\overrightarrow{OM}}{\partial u} \cdot \frac{\partial\overrightarrow{OM}}{\partial v} = \overrightarrow{M}_u \cdot \overrightarrow{M}_v \\ G = \left(\frac{\partial\overrightarrow{OM}}{\partial v}\right)^2 = \overrightarrow{M}_v^2 \end{cases}$$

L'expression s'écrit alors :

$$ds^2 = Edu^2 + Gdv^2 + 2Fdudv \quad (\text{A.1})$$

A.1.2 Deuxième forme fondamentale

En ce point M (Fig. 1.8) il est aussi possible de définir un vecteur normé \vec{n} (A.2) normal à \mathcal{S} . Il se présente comme étant la troisième composante du repère direct formé par $(\overrightarrow{M}_u; \overrightarrow{M}_v; \vec{n})$.

$$\vec{n} = \frac{\overrightarrow{M}_u \wedge \overrightarrow{M}_v}{\|\overrightarrow{M}_u \wedge \overrightarrow{M}_v\|} = \frac{\overrightarrow{M}_u \wedge \overrightarrow{M}_v}{\sqrt{EG - F^2}} \quad (\text{A.2})$$

Soient :

$$\begin{cases} L = \frac{\partial^2\overrightarrow{OM}}{\partial u^2} \cdot \vec{n} \\ N = \frac{\partial^2\overrightarrow{OM}}{\partial v^2} \cdot \vec{n} \\ M = \frac{\partial^2\overrightarrow{OM}}{\partial u \partial v} \cdot \vec{n} = \frac{\partial^2\overrightarrow{OM}}{\partial v \partial u} \cdot \vec{n} \end{cases}$$

La seconde forme fondamentale déterminant la courbure minimale est alors notée :

$$K_n = \frac{Ldu^2 + Ndv^2 + 2Mdudv}{Edu^2 + Gdv^2 + 2Fdudv} \quad (\text{A.3})$$

Le détail de cette formulation est consultable dans [La Greca, 2001].

A.1.3 Courbures principales

De la seconde forme fondamentale se déduit l'expression des deux courbures principales, notées k_1 et k_2 , en tout point M de \mathcal{S} .

Soit :

$$\begin{cases} A = EG - F^2 \\ B = 2FM - EN - LG \\ C = NL - M^2 \end{cases}$$

Les deux courbures principales sont :

$$\begin{cases} k_1 = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \\ k_2 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \end{cases} \quad (\text{A.4})$$

Le détail de cette formulation est consultable dans [La Greca, 2001]. Ces courbures k_1 et k_2 permettent, entre autres, d'apprécier la forme locale de la surface \mathcal{S} en un point M par l'intermédiaire de plusieurs types de courbures qu'elles permettent de calculer.

A.1.4 Courbure gaussienne

La *courbure gaussienne* (A.5) en un point M d'une surface \mathcal{S} est un scalaire noté K qui se calcule par le produit des deux courbures principales (A.4) trouvées en ce point.

$$K = k_1 k_2 = \frac{C}{A} = \frac{NL - M^2}{EG - F^2} \quad (\text{A.5})$$

En étudiant le signe de K , il est possible de déduire des informations sur la surface \mathcal{S} au voisinage du point M . Ainsi, si :

- $K > 0$: Les courbures k_1 et k_2 ont même signe, ainsi M est soit un point *convexe* ($k_1 > 0$ et $k_2 > 0$) soit un point *concave* ($k_1 < 0$ et $k_2 < 0$).
- $K < 0$: Les courbures k_1 et k_2 sont de signes opposés, le point M est alors de type *selle*.
- $K = 0$: Au moins l'une des deux courbures principales est nulle, la forme est localement *cylindrique*. Si la courbure non nulle est positive, la forme est localement *cylindrique convexe*, si elle est négative, elle est localement *cylindrique concave*. Si les deux courbures principales sont nulles, le point M est un *point plat*.

A.1.5 Courbure moyenne

La *courbure moyenne* (A.6) en un point M d'une surface \mathcal{S} est un scalaire noté H qui se calcule par la moyenne des deux courbures principales (A.4) trouvées en ce point.

$$H = \frac{k_1 + k_2}{2} = -\frac{B}{2A} = \frac{LG + EN - 2FM}{2(EG - F^2)} \quad (\text{A.6})$$

Elle permet de préciser les résultats obtenus avec l'étude du signe de la courbure gaussienne et de différencier les sous-cas tels que la convexité ($H > 0$) et la concavité ($H < 0$) pour les points où $K > 0$. Elle est aussi très intéressante, puisqu'elle reste nulle dans le cas des surfaces particulières dites *minimales*.

A.1.6 Courbure absolue

La *courbure absolue* (A.6) en un point M d'une surface \mathcal{S} est un scalaire noté K_{Abs} qui se calcule par la somme des carrés des deux courbures principales (A.4) trouvées en ce point.

$$K_{Abs} = k_1^2 + k_2^2 = \frac{B^2}{A^2} - \frac{2C}{A} = 4H^2 - 2K \quad (\text{A.7})$$

Elle permet de déterminer les zones planes où $K_{Abs} = 0$. Ailleurs, plus la valeur de K_{Abs} est grande, plus la surface est localement «*bombée*».

A.2 Géométrie discrète

A.2.1 Complexe simplicial

Un maillage triangulaire (Fig. 1.3) peut se définir comme un ensemble de simplexes appelé aussi **complexe simplicial** K .

Définition A.2.1 – *Un simplexe ou k -simplexe est l'enveloppe convexe de $k + 1$ points linéairement indépendants de \mathbb{R}^k . De ce fait, un k -simplexe se construit à partir de $k + 1$ $(k - 1)$ -simplexes (Fig. 1.3).*

Le **complexe simplicial** K ainsi défini vérifie la propriété suivante : *deux simplexes de K ne s'intersectent pas du tout, ou s'intersectent selon un simplexe de dimension inférieure qui est leur partie commune de dimension maximale.*

Le terme de *k -complexe* est utilisé pour un complexe simplicial constitué de simplexes de dimension maximale k . Si un k -complexe est constitué de simplexes qui sont tous associés à un k -simplexe alors il est appelé *k -complexe pur* sinon il est appelé *k -complexe non-pur* ce qui est le cas par exemple d'un 2-complexe contenant une arête (1-simplexe) associée à aucune face (2-simplexe) (Fig. A.1).

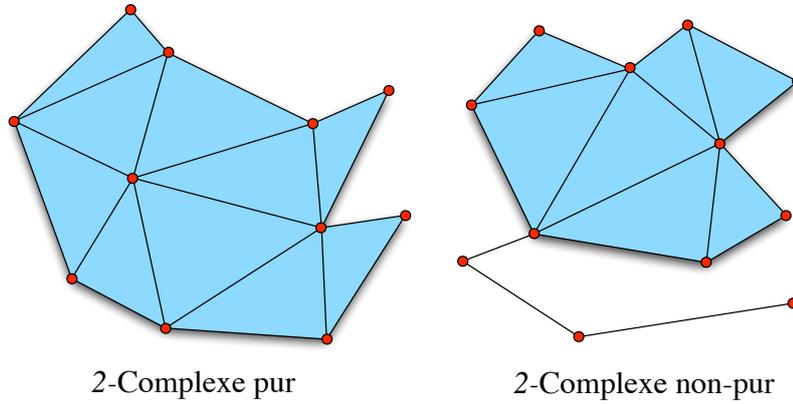


Figure A.1. Complexe simplicial pur et non-pur

A.2.2 Validité de maillage

En utilisant ces définitions, un maillage peut être validé comme étant un solide de topologie correct. Pour cela, il doit, entre autre, vérifier la **relation d'Euler**, répondre aux critères du **défaut angulaire** et être en **2-variété**.

A.2.2.1 La relation d'Euler

Si la **caractéristique d'Euler** χ [Foley and al., 1997] est connue pour l'objet à étudier, le maillage doit vérifier la relation $S - A + F = \chi$ où S est le nombre de sommets, A le nombre d'arêtes et F le nombre de faces. Poincaré précise la caractéristique d'Euler en posant $\chi = 2(1 + T - G) + H$ où T est le nombre de surfaces fermées «*manifold*» (Nombre de vides enfermés dans le solide), G le nombre de trous qui traversent complètement le solide (Genre du solide) et H le nombre de trous dans les faces.

Dans le cas d'un maillage sans bord et formé exclusivement de triangles, la relation $2A = 3F$ est vraie et la relation d'Euler devient $2S - F = 2\chi = 4(1 + T - G) + 2H$;

A.2.2.2 Contrainte de défaut angulaire

Soit $d_i = 2\pi - \sum_i \alpha_i$ le **défaut angulaire** au sommet S_i (Fig. A.2), Alboul et van Damme [Alboul and Van Damme, 1995] posent la condition $\sum_i d_i = 2\pi\chi$ sur le défaut angulaire pour mieux apprécier la cohérence du maillage. Cette condition se démontre à partir de la relation d'Euler.

Remarque : Deux sommets $S_i \in K$ et $S_j \in K$ sont dits **voisins** si l'arête $(S_i, S_j) \in K$. Le **1-voisinage** d'un sommet $S_i \in K$ se définit alors comme étant l'ensemble $V_1(S_i) = \{S_j/S_i, S_j \in K\}$.

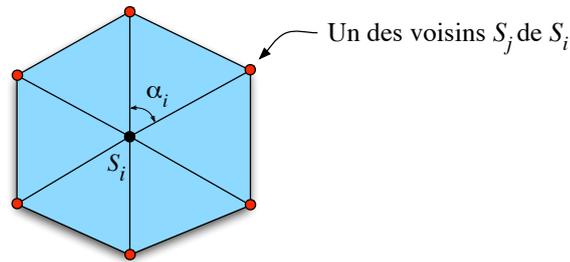


Figure A.2. Voisinage et défaut angulaire d'un sommet S_i

A.2.2.3 Être en 2-variété sans trou ni tétraèdre résiduel

Être en 2-variété (2-manifold) [Foley and al., 1997] revient à dire que tout voisinage d'un point de la frontière est homéomorphe à un disque, c'est-à-dire que le maillage est complet et fermé ou bien que c'est un 2-complexe pur, connexe, orientable et fermé. Cette propriété importante se vérifie si et seulement si une arête est frontière de deux et seulement deux faces, c'est-à-dire si et seulement si :

- Une arête sépare exactement deux faces, le maillage est localement en 2-variété ;
- Une arête est la frontière d'une et une seule face, présence d'un bord : le maillage n'est localement pas en 2-variété, il est donc considéré comme localement non valide ;
- Une arête est le lieu de jonction de plus de deux faces, le maillage est considéré comme non valide puisque qu'il existe un ou plusieurs tétraèdres résiduels.

LA DESCRIPTION : LES PREMIERS PAS D'UNE APPROCHE DÉCLARATIVE

B.1 Première expérience sur la description de surfaces

Cette partie présente le détails des premiers retours de descriptions correspondant à la première expérience (Cf. partie 2.3) :

1. **Professeur des écoles**, formation scientifique, Sexe : féminin
Les parties gauches et supérieures sont bombées (en forme de L renversé). La partie restante forme un rectangle dans lequel on voit un creux en son centre.
2. **Professeur des écoles**, formation littéraire, Sexe : féminin
Une sorte de quadrilatère de 7 cm sur 9 cm dont 2 cotés consécutifs, celui de gauche et du haut, ont reçus un «gonflage» respectivement sur 3 cm et 2 cm de large de telle façon qu'ils sont en forme de coussins. Le reste de la figure n'a pas été gonflé et présente en son milieu un renforcement en forme de cône, le diamètre de ce creux étant en surface de 3 cm.
3. **Informaticien DESS**, Sexe : masculin
Surface à base rectangulaire divisée en 4. Le quart inférieur droit est creusé. Les 3 autres quarts sont bombés en continu le long de leur arête commune.
4. **Infographiste**, Sexe : masculin
 - Plan de type NURBS déformé
 - 4 déformations :
 - 3 bombées et rattachées par une arête de façon à former un angle droit.
 - 1 qui creuse mais qui a l'air moins bombée, plus pointue (moins de points tirés ou déformation adaptative moins forte)
 - Elle est recouverte d'un blinn ou phong
5. **Ingénieur maître Informaticien**, Sexe : masculin
Les parties foncées représentent des creux on peut même parler d'une faille ou canyon central qui longe un espèce de plateau lisse sur la partie gauche. A droite un mamelon taché de ce que je suppose être un rayon ou reflet de lumière. On peut remarquer d'ailleurs que plusieurs sources lumineuses sont présentes. Mise à part cette bosse prononcée toute la partie droite paraît être en pente pour finir au fond dans la zone sombre. Les points culminants seraient donc pour moi en haut de l'image.
6. **Ingénieur maître Informaticien**, Sexe : masculin
On dirait la moitié d'un canapé. Et une personne de forte corpulence vient de se lever.

7. **Pilote de ligne**, Sexe : masculin

Vision de cette surface : Une texture plastique et laquée donc réfléchissant la lumière. Une forme asymétrique qui fait penser à un demi-cylindre en L avec une déformation créée par une boule.

8. **Technicien informaticien**, Sexe : masculin

La surface se compose d'une texture style plastique appliquée sur deux figures une en forme de L aux bords arrondis avec le milieu du L légèrement enflé par rapport aux extrémités, la seconde figure serait un cône Il y a une source de lumière se trouvant environ au sud-est légèrement surélevée par rapport au plan inférieur de la figure Les coins de la texture (du moins 3 visibles) laisse présager que la surface est un découpage dans un ensemble plus grand

9. **DEA Informatique**, Sexe : masculin

Partie gauche en relief avec un relief plus élevé sur la partie basse le tout un peu comme «une colline/un plateau». Partie droite première impression un cône en relief («montagne») et puis bizarrement comme si le relief associé au cône ne pouvait pas exister (par rapport à l'angle de la «montagne» non parallèle/cohérent avec celui de la colline) à ce moment au lieu d'un cône plutôt un «trou/creux».

10. **Illustratrice, peintre**, Sexe : féminin

Pour ma part, cette surface me fait penser à de l'art abstrait. Ce qui importe dans l'art abstrait, c'est l'essence, l'esprit qui libéré de l'objet est saisi par sa forme et sa couleur. Cette surface me rappelle donc des formes anthropomorphes. Le corps d'une jeune femme est mis en avant : la texture est lisse et brillante, il n'y a aucune imperfection, telle que sa peau. Les formes sont sensuelles, on peut distinguer l'aspect d'un sein ferme, une fesse bombée, la cambrure de ses reins, ainsi que ses hanches. La surface est de couleur grise, telle qu'une photo noire et blanc ...

B.2 Deuxième expérience sur la description de formes

Cette section présente les résultats obtenus lors de la réalisation de la deuxième expérience destinée à vérifier qu'il est possible de décrire des formes par la simple donnée d'un texte en français. Ces résultats se divisent en trois parties :

1. les descriptions de formes obtenues et leurs résultats associés ;
2. le vocabulaire utilisé ;
3. des réflexions sur des grammaires de descriptions possibles.

B.2.1 Descriptions de formes obtenues et résultats associés

Dans cette section, nous ne présentons que les descriptions spontanées après leur validation. Ainsi les descriptions qui sont données sont parfois issues de plusieurs cycles de descriptions.

Les descriptions dites «structurées» utilisent la grammaire définie dans la partie B.2.3.2 et le lexique de la partie B.2.2 de ce document.

B.2.1.1 Pièce en «U» (phase 1)



B.2.1.1.1 Description spontanée

- On se place dans un référentiel parallélépipédique défini par les axes x,y,z ;
- On se place dans le plan (x,y) , le regard suivant l'axe z décroissant. L'axe des x est horizontal;
- La pièce est l'agencement en «U» de trois parallélépipèdes : deux suivant l'axe y et un suivant l'axe x . Chaque parallélépipède est trois fois plus long que large et d'épaisseur une fois et demi sa largeur;
- Toutes les arêtes sont arrondies avec le même rayon de raccordement;
- On se place, à présent, dans le plan (x,z) , l'axe x étant horizontal et le regard suivant l'axe des y croissants;
- On rajoute un cylindre d'axe y au centre du rectangle face à nous, de diamètre égal à la largeur de ce même rectangle et de hauteur égale à trois quarts de cette même largeur;
- On perce un trou débouchant au centre du cylindre, de diamètre égal à la moitié du diamètre de ce même cylindre.

B.2.1.1.2 Description structurée

- | | |
|--|---|
| <ul style="list-style-type: none"> - La pièce est dans un repère parallélépipédique d'axe x, y, z; - La pièce est composée de 3 parallélépipèdes; - Ces parallélépipèdes forment un «U»; - Ce «U» est symétrique par rapport à l'axe y; - Les parallélépipèdes sont identiques; - Les parallélépipèdes ont une longueur $L1$; - $L1$ est plus grande que la largeur $L2$; - Les parallélépipèdes ont leur épaisseur plus grande que $L2$ et moins grande que $L1$; | <ul style="list-style-type: none"> - Les parallélépipèdes ont leurs arêtes arrondies; - La pièce est composée d'un cylindre $C1$; - $C1$ a son diamètre $D1$ égal à $L2$; - $C1$ a son axe suivant l'axe y; - $C1$ est placé sur la base du «U»; - $C1$ a un trou $T1$; - $T1$ est débouchant; - $T1$ est centré sur $C1$; - $T1$ a son diamètre égal à la moitié de $D1$. |
|--|---|

Dans la description structurée, il apparaît une précision supplémentaire concernant les trois parallélépipèdes constituant la pièce : ils sont identiques. Cette contrainte supplémentaire a pour conséquence d'écarter le modèle suivant de l'ensemble des solutions possibles :



B.2.1.2 «Cendrier» (phase 1)



B.2.1.2.1 Description spontanée

- Soit un référentiel parallélépipédique défini par les axes x, y, z ;
- L'observateur est situé au-dessus et face au plan (x, y) , son regard est orienté selon l'axe z décroissant, l'axe des x est pour lui l'horizontale;
- On part d'un parallélépipède de base carrée (côté de longueur c) et d'épaisseur $\frac{1}{2}$ de c ;
- Il est positionné de manière à avoir son épaisseur suivant l'axe des z ;
- On se place dans le plan (x, y) ;
- On trace une ligne imaginaire séparant le carré en 2 rectangles identiques;
- Dans la moitié supérieure du carré, il y a une forme creuse, de section rectangulaire. Ce rectangle se trouve à une distance égale au huitième de c de trois des bords du carré et au dixième de c de la ligne imaginaire. Sa largeur est égale à la moitié de sa longueur;
- Dans la moitié inférieure du carré, on imagine une ligne perpendiculaire à la précédente ligne imaginaire séparant en 2 rectangles identiques cette moitié inférieure;
- À gauche de cette nouvelle ligne, il y a une forme creuse de section carrée, se situant au huitième de c des bords du carré initial et au dixième de c des frontières imaginaires;
- On réalise le symétrique de la forme creuse carrée précédente par rapport à la deuxième frontière;
- Toutes les formes creuses sont non débouchantes et de profondeur égale aux trois-quarts de la hauteur du parallélépipède initial;
- Toutes les arêtes sont arrondies.

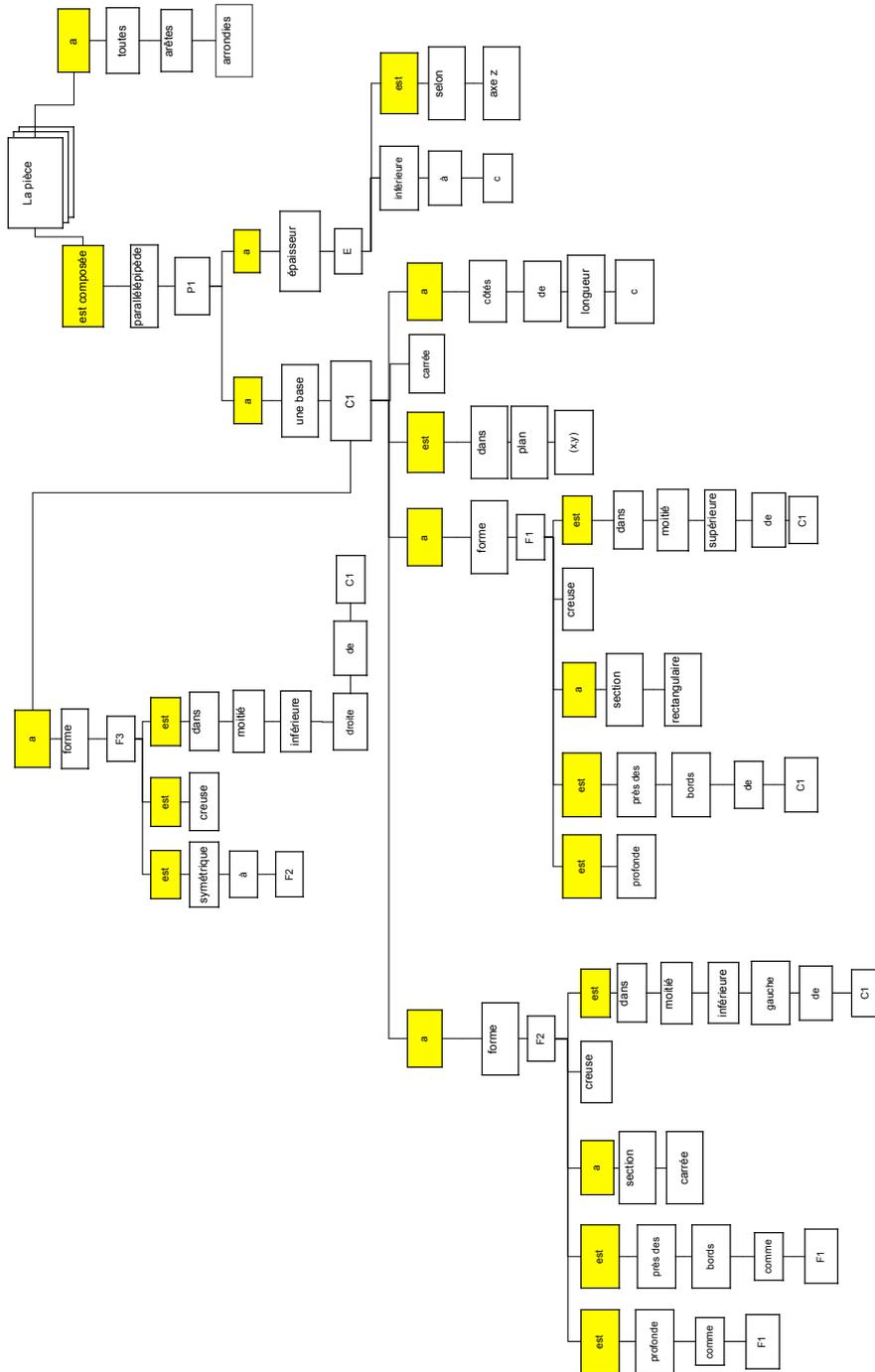
B.2.1.2.2 Description structurée

- | | |
|--|--|
| <ul style="list-style-type: none"> - Soit un référentiel parallélépipédique défini par les axes x, y, z; - La pièce est composée d'un parallélépipède $P1$; - $P1$ a une base B rectangulaire; - B a une longueur L; - B a sa largeur inférieure à L; - $P1$ a son épaisseur inférieure à L; - $P1$ a son épaisseur selon l'axe z; - B a une forme $F1$ creuse; - $F1$ a une section rectangulaire R; - $F1$ se trouve près des bords de B; - $F1$ a une profondeur faible; - R a une forme $F2$ creuse - $F2$ est dans la partie droite de R; - $F2$ a une section carrée C; - $F2$ se trouve près des bords de R; - $F2$ est profonde; | <ul style="list-style-type: none"> - C a un trou T débouchant; - T a son centre près du bord gauche de C; - T a un diamètre petit; - T a un chanfrein sur C; - R a 8 formes bombées dans sa partie gauche; - Ces formes sont identiques; - Ces formes sont centrées dans la partie gauche; - Ces formes sont des rectangles $R2$; - $R2$ a une longueur L; - $R2$ a une largeur l; - L est plus grande que l; - Ces formes ont les angles arrondis; - Ces formes ont une hauteur faible par rapport à l'épaisseur de $P1$; - Ces formes sont parallèles selon l'axe x; - Ces formes sont espacées régulièrement. |
|--|--|

B.2.1.2.3 Résultat



B.2.1.2.4 Organigramme de description



B.2.1.3 L'«évier» (phase 1)



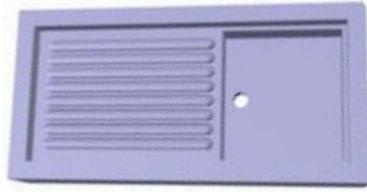
B.2.1.3.1 Description spontanée

- Soit un référentiel parallélépipédique défini par les axes x, y, z ;
- L'observateur est situé au-dessus et face au plan (x, y) , son regard est orienté selon l'axe z décroissant, l'axe des x est pour lui l'horizontale;
- On part d'un parallélépipède de base rectangulaire de longueur l (suivant l'axe x), de largeur environ la moitié de l , et d'épaisseur inférieure à la largeur;
- Il est positionné de manière à avoir son épaisseur suivant l'axe z ;
- On se place dans le plan (x, y) ;
- On fait une première forme creuse de section rectangulaire se situant près de chaque bord du rectangle initial, et de profondeur faible;
- On place une frontière imaginaire parallèle à la largeur, et se situant à $\frac{3}{5}$ de l du bord gauche;
- A droite de la frontière, il y a une forme creuse de section carrée, se situant près des bords de la forme creuse de section rectangulaire, et accolée à la frontière. Sa profondeur est supérieure à la moitié de l'épaisseur du parallélépipède initial;
- On place un trou débouchant sur le fond de cette forme creuse. Son centre se trouve :
 - sur une ligne imaginaire selon l'axe x , coupant la section carrée de la forme creuse en 2 parties égales;
 - près de son bord gauche;
- Le diamètre de ce trou est petit par rapport au côté du carré. Ce trou possède un chanfrein situé sur le fond de la forme creuse de section carrée;
- À gauche de la première frontière imaginaire, il y a 8 formes bombées identiques, centrées dans cet espace;
- Cette forme est un rectangle beaucoup plus long que large dont les angles sont arrondis et dont la hauteur est faible devant l'épaisseur du parallélépipède initial;
- Elles sont parallèles selon l'axe x et équidistantes entre elles.

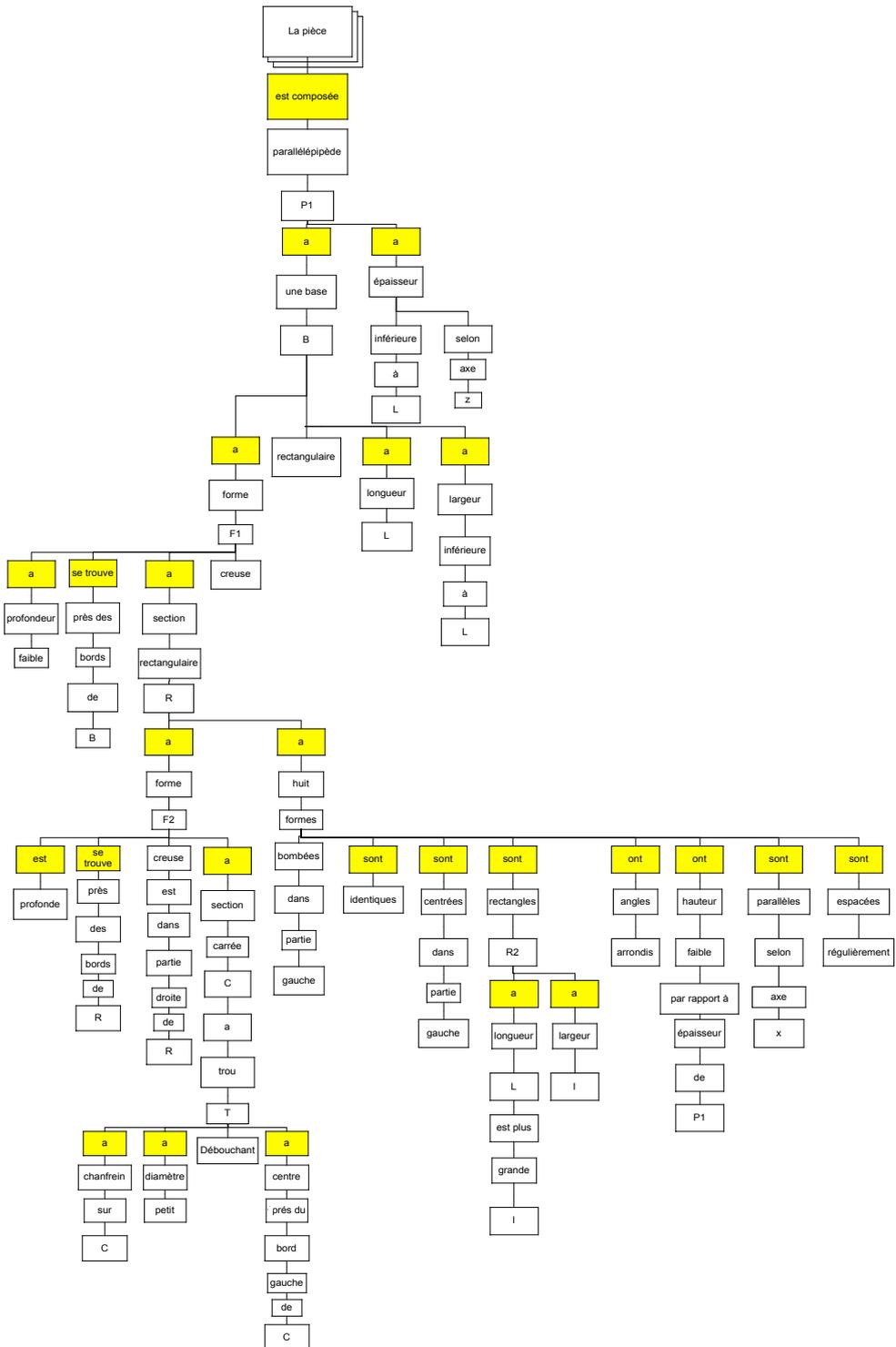
B.2.1.3.2 Description structurée

- | | |
|--|--|
| <ul style="list-style-type: none"> - Soit un référentiel parallélépipédique défini par les axes x, y, z; - La pièce est composée d'un parallélépipède $P1$; - $P1$ a une base B rectangulaire; - B a une longueur L; - B a sa largeur inférieure à L; - $P1$ a son épaisseur inférieure à L; - $P1$ a son épaisseur selon l'axe z; - B a une forme $F1$ creuse; - $F1$ a une section rectangulaire R; - $F1$ se trouve près des bords de B; - $F1$ a une profondeur faible; - R a une forme $F2$ creuse - $F2$ est dans la partie droite de R; - $F2$ a une section carrée C; - $F2$ se trouve près des bords de R; - $F2$ est profonde; | <ul style="list-style-type: none"> - C a un trou T débouchant; - T a son centre près du bord gauche de C; - T a un diamètre petit; - T a un chanfrein sur C; - R a 8 formes bombées dans sa partie gauche; - Ces formes sont identiques; - Ces formes sont centrées dans la partie gauche; - Ces formes sont des rectangles $R2$; - $R2$ a une longueur L; - $R2$ a une largeur l; - L est plus grande que l; - Ces formes ont les angles arrondis; - Ces formes ont une hauteur faible par rapport à l'épaisseur de $P1$; - Ces formes sont parallèles selon l'axe x; - Ces formes sont espacées régulièrement. |
|--|--|

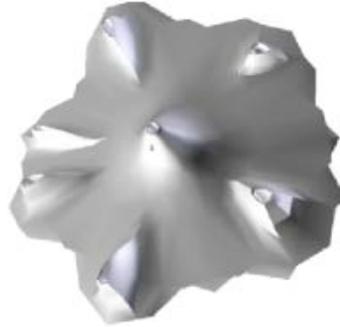
B.2.1.3.3 Résultat



B.2.1.3.4 Organigramme de description



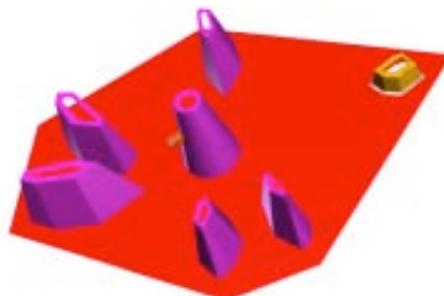
B.2.1.4 L'«île» (phase 2)



B.2.1.4.1 Description structurée

- La pièce est composée d'une surface plane S ;
- S a une section H hexagonale ;
- H a des arêtes de longueurs différentes ;
- Les arêtes sont découpées irrégulièrement ;
- La pièce est composée de 7 formes $F1, F2, F3, F4, F5, F6, F7$;
- Ces formes sont en relief ;
- $F1$ se trouve au centre de H ;
- $F1$ a une forme conique $C1$;
- $C1$ est tronqués ;
- $C1$ a une base $B1$;
- $B1$ se trouve sur H ;
- $F1$ a un creux $Cr1$ au sommet ;
- $Cr1$ est profond par rapport à la hauteur de $C1$;
- $F2, F3, F4, F5, F6, F7$ se trouvent près des sommets de H ;
- $F2, F3, F4, F5, F6, F7$ ont des formes pyramidales $P2, P3, P4, P5, P6, P7$;
- $P2, P3, P4, P5, P6, P7$ sont tronquées ;
- $P2, P3, P4, P5, P6, P7$ ont des bases $B2, B3, B4, B5, B6, B7$;
- $B2, B3, B4, B5, B6, B7$ sont hexagonale $H2, H3, H4, H5, H6, H7$;
- $B2, B3, B4, B5, B6, B7$ sont irrégulières ;
- $H2, H3, H4$ sont plus allongés que $H5, H6, H7$;
- $F2, F3, F4, F5, F6, F7$ ont une hauteur inférieure à $F1$;
- $H7$ est moins effilé que $H2, H3, H4, H5, H6$;
- $H7$ est moins haute que $H2, H3, H4, H5, H6$;
- $F2, F3, F4, F5, F6, F7$ ont un creux $Cr2, Cr3, Cr4, Cr5, Cr6$ au sommet ;
- $Cr2, Cr3, Cr4, Cr5, Cr6$ sont identiques à $H2, H3, H4, H5, H6$;
- $Cr2, Cr3, Cr4, Cr5, Cr6$ sont profonds ;
- $F7$ a un creux $Cr7$ au sommet ;
- $Cr7$ est identique à $H7$;
- $Cr7$ a une profondeur faible ;
- $C1$ a une pyramide $P8$ sur la surface S latérale ;
- $P8$ est identique à $P2, P3, P4, P5, P6, P7$;
- $P8$ a une hauteur h ;
- h est perpendiculaire à S ;
- $P8$ se trouve à mi-hauteur de $F1$;
- La pièce a des arêtes arrondies ;
- H a des arêtes vives.

B.2.1.4.2 Résultat



B.2.1.5 Capot de Twingo® (phase 2)



(a) Vue de face



(b) Vue de coté

B.2.1.5.1 Description structurée

- La pièce se trouve dans un repère parallélépipédique d'axes x, y, z ;
- La pièce est mince dans le plan (x, z) ;
- L'axe x est horizontal vers la droite;
- L'axe z est vertical vers le haut;
- La pièce suit une courbe C dans le plan (x, z) ;
- C a 2 zones;
- La zone A est à gauche;
- La zone A forme une ligne droite inclinée vers le bas et la droite;
- La zone B forme une ligne droite vers le bas et la droite;
- La zone B est inclinée par rapport à la zone A sans angle vif;
- La zone B est inclinée faiblement;
- La zone B est plus petite que la zone A;
- La pièce est plane dans le plan (y, z) ;
- L'axe y est horizontal vers la droite;
- L'axe z est vertical vers le haut;
- La pièce est rectangulaire dans le plan (y, z) ;
- La pièce a une longueur L et une largeur l ;
- L est plus grande que l ;
- La pièce a C pour profil;
- La zone B a 2 trous $T1$ et $T2$;
- $T1$ et $T2$ sont débouchants;
- $T1$ et $T2$ ont une section S ;
- S est un demi-cercle DC ;
- DC a un axe de symétrie AS ;
- AS est selon z ;
- DC a un rayon R ;
- R est plus grand que la largeur de la zone B;
- DC a une base confondue avec la longueur L inférieure;
- $T1$ et $T2$ sont écrasés légèrement;
- $T1$ se trouve à gauche;
- $T2$ se trouve à droite;
- $T1$ est écrasé du côté gauche;
- $T2$ est écrasé du côté droit;
- Les surfaces des zones A et B épousent les trous $T1$ et $T2$ sans discontinuité;
- Les surfaces des zones A et B épousent les trous $T1$ et $T2$ sans arêtes vives;
- La zone A a 3 trous;
- Ces trous sont identiques;
- Ces trous se trouvent en haut à gauche;
- Ces trous sont débouchants;
- Ces trous ont une section $S1$;
- $S1$ forme un parallélogramme;
- $S1$ a un côté plus grand Lp ;
- Lp est parallèle à l ;
- $S1$ a des angles arrondis.

B.2.1.5.2 Résultat



B.2.1.6 Rétroviseur de formule RenaultTM (phase 2)



(a) Vue de face



(b) Vue de côté



(c) Vue arrière

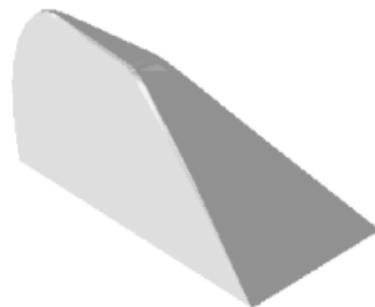
B.2.1.6.1 Description structurée

- La pièce se trouve dans un repère parallélépipédique d'axes x,y,z ;
 - La pièce est composée d'une surface plane P ;
 - P a une section R rectangulaire dans le plan (x, z) ;
 - L'axe x est horizontal vers la droite;
 - L'axe z est vertical vers le haut;
 - La pièce est composée d'une surface bombée S ;
 - S est selon l'axe y croissant;
 - S a pour base R ;
 - S a la courbe $C1$ pour projection dans le plan (x, y) ;
 - L'axe y est vertical vers le haut;
 - L'axe x est horizontal vers la droite;
 - $C1$ est composée de 3 zones A, B, C ;
- $C1$ est sans angle vif;
 - A forme un quart de cercle CL ;
 - CL est vers le haut et la droite;
 - B forme une droite;
 - B est vers le bas et la droite;
 - B est dans le prolongement de CL ;
 - C forme une droite;
 - C est vers le bas et la droite;
 - C est plus petite que B ;
 - S a la courbe $C2$ pour projection dans le plan (y, z) ;
 - L'axe z est vertical vers le haut;
 - L'axe y est horizontal vers la droite;
 - $C2$ forme un triangle T isocèle;
 - T a un sommet SO principal;
 - SO est arrondi.

B.2.1.6.2 Résultat



(a) Vue par dessous



(b) Vue de coté

B.2.2 Détail du vocabulaire utilisé

Le lexique qui est ici proposé regroupe tous les termes ayant été utilisés dans les descriptions structurées qui ont été présentées. La liste n'est donc pas exhaustive mais donne une base des termes que l'on peut trouver dans chaque catégorie.

B.2.2.1 Classification par type de vocabulaire avec nature grammaticale

Localisation					
Terme	Nature	Terme	Nature	Terme	Nature
moitié	nom	être placé sur	verbe	repère	nom
bas	nom	gauche	adjectif	se trouver	verbe
centré	adjectif	gauche	nom	selon	préposition
confondu	adjectif	haut	nom	sur	préposition
dans	préposition	horizontal	adjectif	vers	préposition
droite	adjectif	partie	nom	vertical	adjectif
droite	nom	près	préposition	zone	nom
espacé	adjectif	prolongement	nom		
à côté de	préposition	au dessus de	préposition		

Quantification					
Terme	Nature	Terme	Nature	Terme	Nature
un, deux ...	adjectif	peu*	adverbe	très*	adverbe
demi, tiers ...	nom	faiblement*	adverbe	très très*	adverbe
tout*	adjectif	légèrement	adverbe	extrêmement*	adverbe
grand	adjectif	assez*	adverbe	follement*	adverbe
faible	adjectif	normalement*	adverbe	extraordinairement*	adverbe
petit	adjectif	suffisamment*	adverbe	terriblement*	adverbe
très très peu*	adverbe	passablement*	adverbe	immensément*	adverbe
très peu*	adverbe	moyennement*	adverbe	principalement	adverbe
assez peu*	adverbe	bien*	adverbe		

Les termes comme «*normalement*» ou «*moyennement*» sont des quantificateurs par défaut. Ils peuvent être supprimés de la description sans en changer le sens.

(*) Plusieurs termes présentés dans ce tableau sont tirés de [Desmontils, 1995a].

Contrainte					
Terme	Nature	Terme	Nature	Terme	Nature
identique	adjectif	centré	adjectif	confondu	adjectif
égal	adjectif	symétrique	adjectif	discontinuité	nom
débouchant	adjectif	parallèle	adjectif	isocèle	adjectif
moitié	nom	régulièrement	adverbe		

B.2 DEUXIÈME EXPÉRIENCE SUR LA DESCRIPTION DE FORMES

Géométrie					
Terme	Nature	Terme	Nature	Terme	Nature
pièce	nom	section	nom	plan	adjectif
parallélépipède	nom	bord	nom	profil	nom
axe	nom	rectangulaire	adjectif	demi-cercle	nom
longueur	nom	carré	adjectif	axe de symétrie	nom
largeur	nom	profondeur	nom	rayon	nom
épaisseur	nom	centre	nom	surface	nom
arête	nom	chanfrein	nom	parallélogramme	nom
cylindre	nom	rectangle	nom	discontinuité	nom
diamètre	nom	angle	nom	cercle	nom
base	nom	hauteur	nom	triangle	nom
plan	nom	parallélépipédique	adjectif	sommet	nom
côté	nom	courbe	nom	circulaire	adjectif
forme	nom	ligne	nom	conique	adjectif

Topologie					
Terme	Nature	Terme	Nature	Terme	Nature
trou	nom	troué	adjectif	intérieur	nom
extérieur	nom	intérieur	adjectif	extérieur	adjectif

Morphologie					
Terme	Nature	Terme	Nature	Terme	Nature
profond	adjectif	épouser	verbe	plat	adjectif
incliné	adjectif	vif	adjectif	bosselé	adjectif
creux	adjectif	tronqué	adjectif		
bombé	adjectif	courbe	adjectif		

Jonction						
Terme	Nature		Terme	Nature	Terme	Nature
et	conjonction de coordination		.	ponctuation	,	ponctuation
;	ponctuation					

Comparaison			
Terme	Nature	Terme	Nature
supérieur	adjectif	est excessivement plus ... que*	verbe
est plutôt plus ... que*	verbe	est plutôt un peu plus ... que*	verbe
est légèrement plus ... que*	verbe	est beaucoup plus ... que*	verbe
par rapport à	préposition	est excessivement moins ... que*	verbe
est plutôt moins ... que*	verbe	est plutôt un peu moins ... que*	verbe
comme	préposition	est moins ... que*	verbe
est aussi ... que*	verbe	est un peu plus ... que*	verbe
est légèrement moins ... que*	verbe	est beaucoup moins ... que*	verbe
est plus ... que*	verbe	est un peu moins ... que*	verbe

(*) Plusieurs termes présentés dans ce tableau sont tirés de [Desmontils, 1995a].

Grammaire					
Terme	Nature	Terme	Nature	Terme	Nature
être composé de	verbe de composition	de	préposition	sans	préposition
être	verbe de propriété e	à	préposition	pour	préposition
avoir	verbe de propriété a	du	préposition		

Cette catégorie regroupe les termes permettant d'exprimer la structure de l'objet ou de la forme décrite.

B.2.3 Réflexions sur des grammaires de descriptions possibles

Les grammaires ici présentées ne sont pas des grammaires simplifiées de la langue française. Ce sont simplement les résultats de réflexions entreprises pour la réalisation d'un pseudo-langage basé sur la langue française destiné à la description de formes mécaniques. Ce domaine de recherche n'étant pas le nôtre, ce travail n'a pas été approfondi dans le cadre de cette thèse et reste ouvert sur des pistes de travail.

La notation utilisée pour la présentation des deux grammaires ce-dessous est la notation en «Backus Naur Form»¹ [Naur and al., 1963].

B.2.3.1 Première grammaire de description

<D> ::= <GN> <GV>	<GCCL> ::= <Prep> <GN>
<GN> ::= <GN1>	<GSubRel> ::= <GN> <GV>
<GN1> <Conj> <GN>	<GV>
<GN1> ::= <Nom> <GQual1>	<Gprep> ::= <Prep> <GN>
<Adj> <Nom> <GQual1>	<GCCM> ::= <GPrep> <Adj>
<Nom>	<Adv>
<GQual> ::= <Adj>	<GV> ::= <V> <GN>
<GCCL>	<Ve> <GAttribut>
<GSubRel>	<Ve> <GN>
<Comp> <GN>	<Ve> <Comp> <GN>
<Comp> <Adj> <GN>	<GAttribut> ::= <GCCL>
<Adv> <Comp> <GN>	<Adj>
<Adj> <GCCL>	<Adj> <GN>
<Adj> <GCCM>	<Adj> <Conj> <Adj>
<Adj> <GPrep>	<GV1> ::= <GV>
<GPrep>	<GV> <Conj> <GV1>
<GQual1> ::= <GQual>	
<GQual> <Conj> <GQual1>	

avec

D : Description	GPrep : Groupe prépositionnel
GN : Groupe nominal	GSubRel : Groupe subordonnée relatif
Conj : Conjonction	Adv : Adverbe
GQual : Groupe qualificatif	Prep : Préposition (selon, suivant, dans ...)
Adj : Adjectif quantitatif	Ve : Verbe d'état
GCCL : Groupe comp. circonstanciel de lieu	V : Verbe n'étant pas d'état
Comp : Comparatif	GAttribut : Groupe attribut
GCCM : Groupe comp. circonstanciel de manière	

¹ ::= signifie «est défini par»; | signifie «ou»; <...> est utilisé pour encadrer une catégorie de nom.

Cette grammaire reste beaucoup trop complexe pour les premiers besoins de description à ce stade de la réflexion. Une simplification est donc entreprise et aboutit à une deuxième grammaire.

B.2.3.2 Deuxième grammaire de description

<D>	::=	<GN1> <GN1> <GV> <Conj> <D>
<GN>	::=	<Nom> <Nom> <Ident1> <Nom> <GQual> <Adj> <Nom> <Ident1> <Nom> <GPrep> <Nom> <Ident1> <GQual>
<GN1>	::=	<GN> <GN> <Conj> <GN>
<GQual>	::=	<Adj> <Adj> <Adj> <GPrep> <Adj> <GPrep> <GPrep> <Adj> <Adv>
<GPrep>	::=	<Prep> <GN>
<Ident1>	::=	<Ident> <Ident> <Conj> <Ident>
<GV>	::=	<VCompo> <GN> <Ident1> <VMorpho> <GN> <VPrope> <GQual> <VPropa> <GN> <VCompar> <Adj> <GN> <VLocal> <GPrep>

avec

Ident	:	Identifiant, <i>Ex</i> : «le parallélépipède P1»
Vcompo	:	verbe «de composition», <i>Ex</i> : «est composé de»
Vprope	:	verbe donnant une propriété de dérivation, <i>Ex</i> : «est un»
Vpropa	:	verbe donnant une propriété de caractérisation, <i>Ex</i> : «a», «est»
Vcompar	:	verbe comparatif, <i>Ex</i> : «est plus ... que»
Vlocal	:	verbe de localisation, <i>Ex</i> : «se trouve»

Cette grammaire plus simple permet de réaliser des descriptions plus claires ne faisant apparaître que des phrases courtes. Les descriptions sont plus longues mais leur compréhension plus aisée.

LA SÉMANTIQUE UTILISÉE

C.1 Syntaxe des modèles sémantiques

Les modèles sémantiques proposés sont structurés selon le langage XML. À travers la DTD, la norme XML précise la grammaire syntaxique d'un document XML permettant ainsi de le valider. La DTD que nous proposons à ce jour pour nos modèles sémantiques est la suivante :

```

<!ELEMENT Surface (Localisation?, Forme?,
    Deformation*, Contraintes?, Polygone?,
    Zone*)>
<!ATTLIST Surface Id CDATA #IMPLIED
    Couleur CDATA #IMPLIED>
<!ELEMENT Forme (Propriete*)>
<!ATTLIST Forme Id CDATA #REQUIRED>
<!ELEMENT Deformation (Quantificateur?)>
<!ATTLIST Deformation Id CDATA #REQUIRED>
<!ELEMENT Zone (Localisation?, Forme?,
    Deformation*, Contraintes?, Polygone?,
    Zone*)>
<!ATTLIST Zone Id CDATA #IMPLIED
    Couleur CDATA #IMPLIED>
<!ELEMENT Propriete (Quantificateur?)>
<!ATTLIST Propriete Id CDATA #REQUIRED
    U CDATA #IMPLIED
    V CDATA #IMPLIED>
<!ELEMENT Quantificateur (#PCDATA)>
<!ATTLIST Quantificateur Id CDATA #REQUIRED
    Ecart CDATA #IMPLIED
    Valeur CDATA #IMPLIED
    Poids CDATA #IMPLIED>
<!ELEMENT Localisation (Absolue*, Relative*)>
<!ELEMENT Absolue (Configuration?,
    Quantificateur?)>
<!ATTLIST Absolue Id CDATA #REQUIRED>
<!ELEMENT Relative (Configuration?,
    Quantificateur?, IdZone+)>
<!ATTLIST Relative Id CDATA #REQUIRED>
<!ELEMENT Configuration>
<!ATTLIST Configuration Id CDATA #REQUIRED>
<!ELEMENT IdZone (#PCDATA)>
<!ELEMENT Polygone (VecteurNoeuds?, Sommet+,
    Moins*)>
<!ATTLIST Polygone Degre CDATA #IMPLIED>
<!ELEMENT VecteurNoeuds (N+)>
<!ELEMENT N (#PCDATA)>
<!ELEMENT Sommet (#PCDATA)>
<!ATTLIST Sommet Id CDATA #IMPLIED
    U CDATA #REQUIRED
    V CDATA #REQUIRED
    W CDATA #IMPLIED>
<!ELEMENT Moins (VecteurNoeuds?, Sommet+)>
<!ATTLIST Moins Degre CDATA #IMPLIED
    N CDATA #IMPLIED>
<!ELEMENT Contraintes (Point*, Courbe*)>
<!ELEMENT Courbe (VecteurNoeuds?, Sommet*,
    CourbeGauche)>
<!ATTLIST Courbe Degre CDATA #IMPLIED
    Epsilon CDATA #IMPLIED>
<!ELEMENT CourbeGauche (VecteurNoeuds?, Pole+)>
<!ATTLIST CourbeGauche Degre CDATA #IMPLIED>
<!ELEMENT Pole (#PCDATA)>
<!ATTLIST Pole X CDATA #REQUIRED
    Y CDATA #REQUIRED
    Z CDATA #REQUIRED
    W CDATA #IMPLIED>
<!ELEMENT Point (#PCDATA)>
<!ATTLIST Point U CDATA #IMPLIED
    V CDATA #IMPLIED
    X CDATA #REQUIRED
    Y CDATA #REQUIRED
    Z CDATA #REQUIRED
    Epsilon CDATA #IMPLIED>

```

C.2 Base de connaissances

La DTD associée à la base de connaissances ou «base déclarative» que nous proposons est la suivante :

```
<!ELEMENT BaseDeclarative (Deformation+,
    Localisation+, Forme+, Quantificateur+)>
<!ELEMENT Deformation (Dependance*, Raffinement?)>
<!ATTLIST Deformation
    Id CDATA #REQUIRED
    Quantification CDATA #IMPLIED>
<!ELEMENT Dependance (#PCDATA)>
<!ELEMENT Raffinement (#PCDATA)>
<!ELEMENT Localisation (Configuration+)>
<!ATTLIST Localisation
    Id CDATA #REQUIRED
    Quantification CDATA #IMPLIED>
<!ELEMENT Configuration (Polygone)>
<!ATTLIST Configuration
    Id CDATA #REQUIRED>
<!ELEMENT Polygone (VecteurNoeuds?, Sommet+,
    Moins*)>
<!ATTLIST Polygone Degre CDATA #IMPLIED>
<!ELEMENT VecteurNoeuds (N+)>
<!ELEMENT N (#PCDATA)>
<!ELEMENT Sommet (#PCDATA)>

<!ATTLIST Sommet Id CDATA #IMPLIED
    U CDATA #REQUIRED
    V CDATA #REQUIRED
    W CDATA #IMPLIED>
<!ELEMENT Moins (VecteurNoeuds?, Sommet+)>
<!ATTLIST Moins Degre CDATA #IMPLIED
    N CDATA #IMPLIED>
<!ELEMENT Forme (Polygone?, Propriete*)>
<!ATTLIST Forme Id CDATA #REQUIRED
    Base CDATA #IMPLIED>
<!ELEMENT Propriete (#PCDATA)>
<!ATTLIST Propriete
    Id CDATA #REQUIRED
    U CDATA #REQUIRED
    V CDATA #REQUIRED
    Quantification CDATA #IMPLIED>
<!ELEMENT Quantificateur (#PCDATA)>
<!ATTLIST Quantificateur
    Id CDATA #REQUIRED
    Ecart CDATA #REQUIRED
    Valeur CDATA #REQUIRED>
```

Cette base est menée à évoluer avec les possibilités de l'ensemble des processus de notre approche déclarative. À ce jour elle se présente avec les données suivantes :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE BaseDeclarative SYSTEM "BaseDeclarative.dtd">
<BaseDeclarative>
  <!-- Les Deformations -->
  <Deformation Id="BOMBE" Quantification="1">
    <Dependance>BOMBE</Dependance>
    <Dependance>GONFLE</Dependance>
    <Dependance>BOSSELE</Dependance>
    <Dependance>COURBE</Dependance>
    <Dependance>CREUX</Dependance>
    <Dependance>TROUEE</Dependance>
    <Dependance>NONE</Dependance>
    <Raffinement>10.000000</Raffinement>
  </Deformation>
  <Deformation Id="GONFLE" Quantification="1">
    <Dependance>BOMBE</Dependance>
    <Dependance>GONFLE</Dependance>
    <Dependance>BOSSELE</Dependance>
    <Dependance>COURBE</Dependance>
    <Dependance>CREUX</Dependance>
    <Dependance>TROUEE</Dependance>
    <Dependance>NONE</Dependance>
    <Raffinement>15.000000</Raffinement>
  </Deformation>
  <Deformation Id="BOSSELE" Quantification="1">
    <Dependance>TROUEE</Dependance>
  </Deformation>
  <Dependance>NONE</Dependance>
  <Raffinement>25.000000</Raffinement>
</Deformation>
<Deformation Id="COURBE" Quantification="1">
  <Dependance>GONFLE</Dependance>
  <Dependance>BOMBE</Dependance>
  <Dependance>BOSSELE</Dependance>
  <Dependance>COURBE</Dependance>
  <Dependance>CREUX</Dependance>
  <Dependance>TROUEE</Dependance>
  <Dependance>NONE</Dependance>
  <Raffinement>15.000000</Raffinement>
</Deformation>
<Deformation Id="CREUX" Quantification="1">
  <Dependance>GONFLE</Dependance>
  <Dependance>BOMBE</Dependance>
  <Dependance>BOSSELE</Dependance>
  <Dependance>COURBE</Dependance>
  <Dependance>CREUX</Dependance>
  <Dependance>TROUEE</Dependance>
  <Dependance>NONE</Dependance>
  <Raffinement>10.000000</Raffinement>
</Deformation>
<Deformation Id="TROUEE">
  <Dependance>NONE</Dependance>
</Deformation>
```

C.2 BASE DE CONNAISSANCES

```

<Deformation Id="NONE">
</Deformation>

<!-- Les Localisations -->
<Localisation Id="ANGLE HG">
  <Configuration Id="RECTANGULAIRE">
    <Polygone>
      <Sommet Id="S0" U="0" V="50"/>
      <Sommet Id="S1" U="0" V="100"/>
      <Sommet Id="S2" U="50" V="100"/>
      <Sommet Id="S3" U="50" V="50"/>
    </Polygone>
  </Configuration>
  <Configuration Id="TRIANGULAIRE">
    <Polygone>
      <Sommet Id="S0" U="0" V="50"/>
      <Sommet Id="S1" U="0" V="100"/>
      <Sommet Id="S2" U="50" V="100"/>
    </Polygone>
  </Configuration>
  <Configuration Id="RONDE">
    <Polygone Degre="2">
      <VecteurNoeuds>
        <N>0</N><N>0</N><N>0</N>
        <N>25</N><N>25</N>
        <N>75</N><N>75</N>
        <N>100</N><N>100</N><N>100</N>
      </VecteurNoeuds>
      <Sommet Id="S0" U="0" V="50" W="1"/>
      <Sommet Id="S1" U="0" V="100" W="1"/>
      <Sommet Id="S2" U="0" V="100" W="1"/>
      <Sommet Id="S3" U="50" V="100" W="1"/>
      <Sommet Id="S4" U="50" V="100" W="1"/>
      <Sommet Id="S5" U="50" V="50" W="0.7071"/>
      <Sommet Id="S6" U="0" V="50" W="1"/>
    </Polygone>
  </Configuration>
</Localisation>

...

<!-- Les Formes de base -->
<Forme Id="TOTALE"/>
<Forme Id="QUELCONQUE">
  <Propriete Id="LARGE" U="2.000" V="1.000"
    Quantification="1"/>
  <Propriete Id="LONGUE" U="1.000" V="2.000"
    Quantification="1"/>
  <Propriete Id="EQUILIBREE" U="1.000" V="1.000"/>
</Forme>
<Forme Id="CARREE">
  <Polygone>
    <Sommet Id="S0" U="0" V="0"/>
    <Sommet Id="S1" U="0" V="100"/>
    <Sommet Id="S2" U="100" V="100"/>
    <Sommet Id="S3" U="100" V="0"/>
  </Polygone>
</Forme>

<Forme Id="RONDE">
  <Polygone Degre="2">
    <VecteurNoeuds>
      <N>0</N><N>0</N><N>0</N>
      <N>25</N><N>25</N>
      <N>50</N><N>50</N>
      <N>75</N><N>75</N>
      <N>100</N><N>100</N><N>100</N>
    </VecteurNoeuds>
    <Sommet Id="S0" U="0.5" V="0" W="1"/>
    <Sommet Id="S1" U="0" V="0" W="0.7071"/>
    <Sommet Id="S2" U="0" V="0.5" W="1"/>
    <Sommet Id="S3" U="0" V="1" W="0.7071"/>
    <Sommet Id="S4" U="0.5" V="1" W="1"/>
    <Sommet Id="S5" U="1" V="1" W="0.7071"/>
    <Sommet Id="S6" U="1" V="0.5" W="1"/>
    <Sommet Id="S7" U="1" V="0" W="0.7071"/>
    <Sommet Id="S8" U="0.5" V="0" W="1"/>
  </Polygone>
</Forme>
<Forme Id="OVALE" Base="RONDE">
  <Propriete Id="LARGE" U="2.000" V="1.000"
    Quantification="1"/>
  <Propriete Id="LONGUE" U="1.000" V="2.000"
    Quantification="1"/>
</Forme>
<Forme Id="RECTANGULAIRE" Base="CARREE">
  <Propriete Id="LARGE" U="2.000" V="1.000"
    Quantification="1"/>
  <Propriete Id="LONGUE" U="1.000" V="2.000"
    Quantification="1"/>
</Forme>

<!-- Les Quantificateurs -->
<Quantificateur Id="LEGEREMENT"
  Ecart="4.67"
  Valeur="6.000"/>
<Quantificateur Id="FAIBLEMENT"
  Ecart="4.67"
  Valeur="20.000"/>
<Quantificateur Id="PEU"
  Ecart="4.67"
  Valeur="34.000"/>
<Quantificateur Id="MOYENNEMENT"
  Ecart="5.34"
  Valeur="50.000"/>
<Quantificateur Id="ASSEZ"
  Ecart="4.67"
  Valeur="66.000"/>
<Quantificateur Id="TRES"
  Ecart="4.67"
  Valeur="80.000"/>
<Quantificateur Id="ENORMEMENT"
  Ecart="4.67"
  Valeur="94.000"/>
</BaseDeclarative>

```

C.3 Un modèle sémantique généralisé pour le capot de Twingo®

Cette partie présente un modèle sémantique généralisé possible associé à la première description donnée dans la partie 5.3.1 du chapitre des résultats.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Surface SYSTEM "ModeleSemantique.dtd">
<Surface Id="Z0" Couleur="Noir">
  <Localisation>
    <Absolue Id="TOTALE"/>
  </Localisation>
  <Forme Id="QUELCONQUE">
    <Propriete Id="LARGE">
      <Quantificateur Id="EXTREMEMENT PEU"
        Valeur="6" Ecart="4.67" Poids="0"/>
    </Propriete>
  </Forme>
  <Deformation Id="NONE"/>
  <Polygone Degre="2">
    <VecteurNoeuds>
      <N>0</N><N>0</N><N>0</N><N>5.26</N><N>10.53</N>
      <N>15.79</N><N>21.05</N><N>26.32</N><N>31.58</N>
      <N>36.84</N><N>42.10</N><N>47.37</N><N>52.63</N>
      <N>57.89</N><N>63.16</N><N>68.42</N><N>73.68</N>
      <N>78.95</N><N>84.21</N><N>89.47</N><N>94.74</N>
      <N>100</N><N>100</N><N>100</N>
    </VecteurNoeuds>
    <Sommet Id="S0" U="0" V="0" W="1"/>
    <Sommet Id="S1" U="0" V="100" W="1"/>
    <Sommet Id="S1" U="0" V="100" W="1"/>
    <Sommet Id="S2" U="50" V="90" W="1"/>
    <Sommet Id="S3" U="100" V="100" W="1"/>
    <Sommet Id="S3" U="100" V="100" W="1"/>
    <Sommet Id="S4" U="100" V="0" W="1"/>
    <Sommet Id="S4" U="100" V="0" W="1"/>
    <Sommet Id="S5" U="95" V="0" W="1"/>
    <Sommet Id="S5" U="95" V="0" W="1"/>
    <Sommet Id="S6" U="95" V="12" W="0.7071"/>
    <Sommet Id="S7" U="82" V="12" W="0.7071"/>
    <Sommet Id="S8" U="82" V="0" W="1"/>
    <Sommet Id="S8" U="82" V="0" W="1"/>
    <Sommet Id="S9" U="18" V="0" W="1"/>
    <Sommet Id="S9" U="18" V="0" W="1"/>
    <Sommet Id="S10" U="18" V="12" W="0.7071"/>
    <Sommet Id="S11" U="5" V="12" W="0.7071"/>
    <Sommet Id="S12" U="5" V="0" W="1"/>
    <Sommet Id="S12" U="5" V="0" W="1"/>
    <Sommet Id="S0" U="0" V="0" W="1"/>
  </Polygone>
  <Zone Id="Z1" Couleur="Rouge">
    <Localisation>
      <Absolue Id="PARTIE BASSE">
        <Quantificateur Id="GRANDE" Valeur="66"
          Ecart="4.67" Poids="1"/>
      </Absolue>
    </Localisation>
    <Forme Id="TOTALE"/>
    <Deformation Id="COURBE" Param="30">
      <Quantificateur Id="TRES PEU" Valeur="20"
        Ecart="4.67" Poids="0"/>
    </Deformation>
    <Polygone>
      <Sommet Id="S0" U="0" V="97"/>
      <Sommet Id="S1" U="0" V="0"/>
      <Sommet Id="S2" U="100" V="0"/>
      <Sommet Id="S3" U="100" V="97"/>
    </Polygone>
    <Zone Id="Z2" Couleur="Bleu">
      <Localisation>
        <Absolue Id="BANDE HORIZINTALE INFERIEURE">
          <Quantificateur Id="TRES" Valeur="66"
            Ecart="4.67" Poids="0"/>
        </Absolue>
      </Localisation>
      <Forme Id="TOTALE"/>
      <Deformation Id="COURBE" Param="30">
        <Quantificateur Id="PEU" Valeur="34"
          Ecart="4.67" Poids="0"/>
      </Deformation>
      <Polygone>
        <Sommet Id="S0" U="0" V="15"/>
        <Sommet Id="S1" U="0" V="0"/>
        <Sommet Id="S2" U="100" V="0"/>
        <Sommet Id="S3" U="100" V="15"/>
      </Polygone>
      <Zone Id="Z5" Couleur="Jaune">
        <Localisation>
          <Absolue Id="MILIEU"/>
        </Localisation>
        <Forme Id="QUELCONQUE">
          <Propriete Id="LONGUE">
            <Quantificateur Id="PEU" Valeur="34"
              Ecart="4.67" Poids="0"/>
          </Propriete>
        </Forme>
      </Zone>
    </Zone>
  </Surface>

```

Suite ->

```

<Deformation Id="BOMBE" Param="0.5">
  <Quantificateur Id="EXTREMEMENT PEU"
    Valeur="6" Ecart="4.67" Poids="1"/>
</Deformation>
<Polygone>
  <Sommet Id="S0" U="50" V="60"/>
  <Sommet Id="S1" U="52" V="40"/>
  <Sommet Id="S2" U="50" V="20"/>
  <Sommet Id="S3" U="48" V="40"/>
<Moins>
  <Sommet Id="S0" U="50" V="70"/>
  <Sommet Id="S1" U="60" V="50"/>
  <Sommet Id="S2" U="50" V="30"/>
  <Sommet Id="S3" U="40" V="50"/>
</Moins>
</Polygone>
</Zone>
</Zone>
<Zone Id="Z3" Couleur="Vert">
  <Localisation>
    <Absolue Id="BANDE VERTICALE GAUCHE"/>
  </Localisation>
  <Forme Id="CARREE"/>
  <Deformation Id="BOMBE" Param="2">
    <Quantificateur Id="EXTREMEMENT PEU"
      Valeur="6" Ecart="4.67" Poids="0"/>
  </Deformation>
  <Polygone>
    <Sommet Id="S0" U="5" V="13.5"/>
    <Sommet Id="S1" U="5" V="0"/>
    <Sommet Id="S2" U="18" V="0"/>
    <Sommet Id="S3" U="18" V="13.5"/>
  </Polygone>
</Zone>
<Zone Id="Z4" Couleur="Vert">
  <Localisation>
    <Absolue Id="BANDE VERTICALE DROITE"/>
  </Localisation>
  <Forme Id="CARREE"/>
  <Deformation Id="BOMBE" Param="2">
    <Quantificateur Id="EXTREMEMENT PEU"
      Valeur="6" Ecart="4.67" Poids="0"/>
  </Deformation>
  <Polygone>
    <Sommet Id="S0" U="82" V="13.5"/>
    <Sommet Id="S1" U="82" V="0"/>
    <Sommet Id="S2" U="95" V="0"/>
    <Sommet Id="S3" U="95" V="13.5"/>
  </Polygone>
</Zone>
<Zone Id="Z6" Couleur="Cyan">
  <Localisation>
    <Absolue Id="HAUT-GAUCHE"
      Configuration="RECTANGULAIRE"/>
  </Localisation>
  <Forme Id="QUELCONQUE">

```

Suite ->

```

  <Propriete Id="LONGUE">
    <Quantificateur Id="PEU" Valeur="34"
      Ecart="4.67" Poids="0"/>
  </Propriete>
</Forme>
<Deformation Id="TROUEE"/>
<Polygone Degre="2">
  <VecteurNoeuds>
    <N>0</N><N>0</N><N>0</N><N>25</N><N>25</N>
    <N>50</N><N>50</N><N>75</N><N>75</N>
    <N>100</N><N>100</N><N>100</N>
  </VecteurNoeuds>
  <Sommet Id="S0" U="12" V="92" W="1"/>
  <Sommet Id="S1" U="9" V="92" W="2"/>
  <Sommet Id="S2" U="9" V="84" W="1"/>
  <Sommet Id="S3" U="9" V="76" W="2"/>
  <Sommet Id="S4" U="12" V="76" W="1"/>
  <Sommet Id="S5" U="15" V="76" W="2"/>
  <Sommet Id="S6" U="15" V="84" W="1"/>
  <Sommet Id="S7" U="15" V="92" W="2"/>
  <Sommet Id="S8" U="12" V="92" W="1"/>
</Polygone>
</Zone>
<Zone Id="Z7" Couleur="Cyan">
  <Localisation>
    <Absolue Id="HAUT-GAUCHE"
      Configuration="RECTANGULAIRE"/>
  </Localisation>
  <Forme Id="QUELCONQUE">
  <Propriete Id="LONGUE">
    <Quantificateur Id="PEU" Valeur="34"
      Ecart="4.67" Poids="0"/>
  </Propriete>
</Forme>
<Deformation Id="TROUEE"/>
<Polygone Degre="2">
  <VecteurNoeuds>
    <N>0</N><N>0</N><N>0</N><N>25</N><N>25</N>
    <N>50</N><N>50</N><N>75</N><N>75</N>
    <N>100</N><N>100</N><N>100</N>
  </VecteurNoeuds>
  <Sommet Id="S0" U="20" V="90" W="1"/>
  <Sommet Id="S1" U="17" V="90" W="2"/>
  <Sommet Id="S2" U="17" V="82" W="1"/>
  <Sommet Id="S3" U="17" V="74" W="2"/>
  <Sommet Id="S4" U="20" V="74" W="1"/>
  <Sommet Id="S5" U="23" V="74" W="2"/>
  <Sommet Id="S6" U="23" V="82" W="1"/>
  <Sommet Id="S7" U="23" V="90" W="2"/>
  <Sommet Id="S8" U="20" V="90" W="1"/>
</Polygone>
</Zone>
<Zone Id="Z8" Couleur="Cyan">
  <Localisation>
    <Absolue Id="HAUT-GAUCHE"
      Configuration="RECTANGULAIRE"/>
  </Localisation>

```

```

<Forme Id="QUELCONQUE">
  <Propriete Id="LONGUE">
    <Quantificateur Id="PEU" Valeur="34"
      Ecart="4.67" Poids="0"/>
  </Propriete>
</Forme>
<Deformation Id="TROUÉE"/>
<Polygone Degre="2">
  <VecteurNoeuds>
    <N>0</N><N>0</N><N>0</N><N>25</N>
    <N>25</N><N>50</N><N>50</N><N>75</N>
    <N>75</N><N>100</N><N>100</N><N>100</N>
  </VecteurNoeuds>

```

Suite ->

```

  <Sommet Id="S0" U="28" V="88" W="1"/>
  <Sommet Id="S1" U="25" V="88" W="2"/>
  <Sommet Id="S2" U="25" V="80" W="1"/>
  <Sommet Id="S3" U="25" V="72" W="2"/>
  <Sommet Id="S4" U="28" V="72" W="1"/>
  <Sommet Id="S5" U="31" V="72" W="2"/>
  <Sommet Id="S6" U="31" V="80" W="1"/>
  <Sommet Id="S7" U="31" V="88" W="2"/>
  <Sommet Id="S8" U="28" V="88" W="1"/>
</Polygone>
</Zone>
</Zone>
</Surface>

```

ALGORITHMES PROPOSÉS

D.1 Algorithme de raffinement du polyèdre de contrôle

Cet algorithme de raffinement du polyèdre de contrôle est illustré dans la partie 4.3.3. Il permet d'ajouter des pôles au polyèdre de contrôle au niveau des zones paramétriques devant être déformées. Pour cela, nous utilisons la méthode de raffinement d'un vecteur de nœud de [Boehm and Prautzsch, 1985].

Algorithme D.1 - Raffinement du polyèdre de contrôle

Prérequis :

- Soit U et V les vecteurs de nœuds dans les directions u et v ;
- Soit B la boîte englobante $\{(uMin, vMin); (uMax, vMax)\}$ de la zone Z définie dans le repère global \mathcal{R}_0 ;
- Soit R le seuil de raffinement minimum (issu de la base de connaissances) attaché aux déformations de Z . R est adapté à l'échelle du repère local de Z en rapport avec le repère global \mathcal{R}_0 ;
- Soit L_u (respectivement L_v) la liste ordonnée des nœuds à insérer dans U (respectivement V) ;

- 1: $L_u \leftarrow \emptyset$ {Construction de L_u en fonction de U }
 - 2: **si** les nœuds B_{uMin} et B_{uMax} n'existent pas dans U **alors**
 - 3: Les ajouter dans L_u ;
 - 4: **fin si**
 - 5: **tant que** Il existe deux nœuds successifs $\{U_i, U_{i+1}\}$ dans $\{L_u \cup [uMin, uMax]\}$ avec $U_{i+1} - U_i > R$ **faire**
 - 6: Ajouter à L_u le nœud $\frac{(U_i + U_{i+1})}{2}$;
 - 7: **fin tant que**
 - 8: Construire L_v en fonction de V de la même façon que L_u en fonction de U ;
 - 9: Appliquer la méthode de raffinement de [Boehm and Prautzsch, 1985] sur le vecteur de nœuds U en utilisant L_u et sur le vecteur de nœuds V en utilisant L_v .
-

D.2 Satisfaction de N contraintes ponctuelles sur une courbe NURBS selon la méthode 1 dite globale

Algorithme D.2 - Satisfaction de N contraintes ponctuelles sur une courbe NURBS selon la méthode 1 dite globale

Prérequis : {déclarations}

- V : vecteur de 3 scalaires ;
- m : vecteur de n vecteurs de 3 scalaires ;

- 1: $t \leftarrow 0$ {initialisation du compteur d'itérations}
 - 2: $\varepsilon \leftarrow \sum_{\lambda=0}^{N-1} \|\hat{M}_\lambda - \mathcal{C}(u_\lambda)\|_2$ {initialisation de l'erreur globale courante}
 - 3: Initialisation du vecteur déplacement m_i à $\vec{0}$
 - 4: Initialisation des vecteurs $B(\lambda)$
 - 5: **répéter**
 - 6: {calcul des vecteurs déplacement}
 - 7: **pour** $\lambda = 0$ to $N - 1$ **faire**
 - 8: $V \leftarrow \hat{M}_\lambda - \mathcal{C}(u_\lambda)$ {erreur à corriger}
 - 9: **pour** $i = 0$ to $n - 1$ **faire**
 - 10: $m_i \leftarrow m_i + B_i(\lambda)V$
 - 11: **fin pour**
 - 12: **fin pour**
 - 13: {mise à jour du polygone de contrôle}
 - 14: **pour** $i = 0$ to $n - 1$ **faire**
 - 15: $P_i \leftarrow P_i + \frac{m_i}{N}$
 - 16: **fin pour**
 - 17: $t \leftarrow t + 1$ {mise à jour du compteur d'itérations}
 - 18: {mise à jour de l'erreur globale}
 - 19: *dernier* $\varepsilon \leftarrow \varepsilon$
 - 20: $\varepsilon \leftarrow \sum_{\lambda=0}^{N-1} \|\hat{M}_\lambda - \mathcal{C}(u_\lambda)\|_2$
 - 21: **jusqu'à** ($| \textit{dernier}\varepsilon - \varepsilon | < \delta E$)
-

D.3 Satisfaction de N contraintes ponctuelles sur une courbe NURBS selon la méthode 2 dite séquentielle

Algorithme D.3 - Satisfaction de N contraintes ponctuelles sur une courbe NURBS selon la méthode 2 dite séquentielle

Prérequis : {déclarations}

- V : vecteur de 3 éléments ;
- D : vecteur of N éléments : pondération de la méthode ;

- 1: $t \leftarrow 0$ {initialisation du compteur d'itérations}
- 2: $\varepsilon \leftarrow \sum_{\lambda=0}^{N-1} \|\hat{M}_\lambda - \mathcal{C}(u_\lambda)\|_2$ {initialisation de l'erreur globale courante}
- 3: {initialisation des poids}
- 4: **pour** $\lambda = 0$ to $N - 1$ **faire**
- 5: $Q_\lambda \leftarrow \sum_{i=0}^{n-1} R_i(u_\lambda) B_i(u_\lambda)$
- 6: **fin pour**
- 7: **répéter**
- 8: {correction des contraintes une à une}
- 9: **pour** $\lambda = 0$ to $N - 1$ **faire**
- 10: $V \leftarrow \hat{M}_\lambda - \mathcal{C}(u_\lambda)$ {erreur à corriger}
- 11: {mise à jour du polygone de contrôle}
- 12: **pour** $i = 0$ to $n - 1$ **faire**
- 13: $P_i \leftarrow P_i + \frac{B_i(u_\lambda)}{Q_\lambda} V$
- 14: **fin pour**
- 15: **fin pour**
- 16: $t \leftarrow t + 1$ {mise à jour du compteur d'itérations}
- 17: {mise à jour de l'erreur globale courante}
- 18: *dernier* $\varepsilon \leftarrow \varepsilon$
- 19: $\varepsilon \leftarrow \sum_{\lambda=0}^{N-1} \|\hat{M}_\lambda - \mathcal{C}(u_\lambda)\|_2$
- 20: **si** *dernier* $\varepsilon - \varepsilon <= 0$ **alors**
- 21: Le système est surcontraint : arrêt de la boucle ;
- 22: **fin si**
- 23: **jusqu'à** (*dernier* $\varepsilon - \varepsilon < \delta E$)

PREUVES ET DÉMONSTRATIONS

Une grande partie de ces démonstrations ont été réalisées avec la collaboration d’Alexandra Bac que nous remercions tout particulièrement.

E.1 Preuve de convergence de la méthode globale pour l’influence naturelle des contraintes

Dans cette partie est donnée la preuve de convergence de la méthode globale, appliquée aux courbes, de satisfaction de N contraintes ponctuelles de passage dans le cas d’une influence *naturelle* pour chaque contrainte.

Preuve 1 : Preuve de la proposition 4.3.6

Comme $\sum_{l=1}^t S^{l-1}$ est une série géométrique, il peut être montré [Ciarlet, 1998] que celle-ci converge si et seulement si $\varrho(S) < 1$ où $\varrho(S)$ est le rayon spectral¹ de S .

Lemme E.1.1 : Si $\varrho(S)$ est le rayon spectral de S , la relation suivante est vraie :

$$\max_{x \neq 0} \frac{|\langle Sx, x \rangle|}{\|x\|_2^2} \leq \varrho(S) \quad \forall x \in \mathbb{R}^N \quad (\text{E.1})$$

Preuve :

Lemme E.1.2 : S est diagonalisable lorsque les contraintes ont une influence *naturelle*.

Preuve :

Une contrainte de localisation *naturelle* correspond à l’égalité $\overrightarrow{B(\lambda)} = \overrightarrow{R(\lambda)}$, ce qui rend S symétrique. Or toute matrice symétrique est diagonalisable.

□

S étant diagonalisable, considérons alors V comme l’espace vectoriel construit sur la base

¹la plus grande valeur propre de S en valeur absolue.

orthogonale d_0, \dots, d_{N-1} des valeurs propres de S . Ainsi, $\forall x \in V$, $x = \sum_{i=0}^{N-1} x_i d_i$ avec $\lambda_0, \dots, \lambda_{N-1}$ les valeurs propres de S . Nous avons donc :

$$\begin{aligned} \langle Sx, x \rangle &= \left\langle S \sum_{i=0}^{N-1} x_i d_i, \sum_{i=0}^{N-1} x_i d_i \right\rangle = \left\langle \sum_{i=0}^{N-1} Sx_i d_i, \sum_{i=0}^{N-1} x_i d_i \right\rangle \\ &= \left\langle \sum_{i=0}^{N-1} \lambda_i x_i d_i, \sum_{i=0}^{N-1} x_i d_i \right\rangle = \sum_{i=0}^{N-1} \lambda_i x_i^2 \\ &\text{puisque } \langle d_i, d_j \rangle = 0 \text{ if } i \neq j \\ |\langle Sx, x \rangle| &= \left| \sum_{i=0}^{N-1} \lambda_i x_i^2 \right| \leq \sum_{i=0}^{N-1} |\lambda_i| x_i^2 \text{ and } |\lambda_i| \leq \varrho(S) \\ &\leq \varrho(S) \sum_{i=0}^{N-1} x_i^2 \leq \varrho(S) \|x\|_2^2 \end{aligned}$$

□

Lemme E.1.3 : La limite maximale de l'équation (E.1) est atteinte.

Preuve :

Soit x_0 le vecteur propre associé à la valeur propre λ_0 correspondant à $\varrho(S)$, nous avons :

$$|\langle Sx_0, x_0 \rangle| = |\langle \lambda_0 x_0, x_0 \rangle| = |\lambda_0| \|x_0\|_2^2 = \varrho(S) \|x_0\|_2^2$$

C'est-à-dire :

$$\frac{|\langle Sx_0, x_0 \rangle|}{\|x_0\|_2^2} \tag{E.2}$$

□

Corollaire E.1.1 : En utilisant (E.1) et (E.2), on peut déduire que le rayon spectral de S est :

$$\varrho(S) = \max_{x \neq 0} \frac{|\langle Sx, x \rangle|}{\|x\|_2^2} \tag{E.3}$$

La condition de convergence devient alors : $\frac{|\langle Sx, x \rangle|}{\|x\|_2^2} < 1$ c'est-à-dire $\frac{|\langle (I - \frac{1}{N}D)x, x \rangle|}{\|x\|_2^2} < 1$ ce qui est équivalent à prouver :

$$0 < \frac{\langle Dx, x \rangle}{N\|x\|_2^2} < 2 \tag{E.4}$$

Soit Q une matrice $N \times n$ définie par $\overrightarrow{R(\lambda)}$ sur chaque ligne :

$$Q^T = \left(\overrightarrow{R(0)} \quad \overrightarrow{R(1)} \quad \dots \quad \overrightarrow{R(N-1)} \right)$$

Soit \tilde{Q} une matrice $N \times n$ définie par $\overrightarrow{B(\lambda)}$ sur chaque ligne :

$$\tilde{Q}^T = \left(\overrightarrow{B(0)} \quad \overrightarrow{B(1)} \quad \dots \quad \overrightarrow{B(N-1)} \right)$$

Remarque E.1.1 – Selon les propriétés connues de $\overrightarrow{R(\lambda)}$ and $\overrightarrow{B(\lambda)}$ citées dans le paragraphe 4.3.6.1.1, il est possible de déduire les relations suivantes :

$$\|Q\|_\infty = \|\tilde{Q}\|_\infty = 1, \|Q\|_1 < N \text{ et } \|\tilde{Q}\|_1 < N$$

Lemme E.1.4 : $D = Q\tilde{Q}^T$ est une matrice inversible pour l'influence *naturelle* des contraintes.

Preuve :

Une contrainte de localisation *naturelle* correspond à l'égalité $\overrightarrow{B(\lambda)} = \overrightarrow{R(\lambda)}$, de ce fait $D = Q\tilde{Q}^T$ est symétrique définie positive. Or toute matrice symétrique définie positive est inversible.

□

La condition de convergence (E.4) devient donc :

$$0 < \frac{\langle Q\tilde{Q}^T x, x \rangle}{N\|x\|_2^2} < 2 \Leftrightarrow 0 < \frac{\langle \tilde{Q}^T x, Q^T x \rangle}{N\|x\|_2^2} < 2 \quad (\text{E.5})$$

Lemme E.1.5 :

$$0 < \frac{\langle \tilde{Q}^T x, Q^T x \rangle}{N\|x\|_2^2} \quad \forall x \in \mathbb{R}^N \quad (\text{E.6})$$

Preuve :

Si $\overrightarrow{B(\lambda)} = \overrightarrow{R(\lambda)}$ alors $\langle \tilde{Q}^T x, Q^T x \rangle = \langle Q^T x, Q^T x \rangle = \|Q^T x\|_2^2 > 0$ pour tout $x \neq 0$.

□

Lemme E.1.6 :

$$\frac{\|x\|_1}{\|x\|_2} \leq \sqrt{N} \quad \forall x \in \mathbb{R}^N \quad (\text{E.7})$$

Preuve :

Soit g une fonction telle que $g : \mathbb{R}^N \mapsto \mathbb{R} / g(x) = \frac{\|x\|_1}{\|x\|_2} = \frac{\sum_{i=0}^{N-1} |x_i|}{\sqrt{\sum_{i=0}^{N-1} x_i^2}}, \forall x \in \mathbb{R}^N$

La fonction g atteint un maximum ou un minimum lorsque $\frac{\partial g(x)}{\partial x_\lambda} = 0, \forall \lambda \in [0, N-1]$.

$$\text{et } \frac{\partial g(x)}{\partial x_\lambda} = \begin{cases} \frac{1}{\|x\|_2^3} \left(\sum_{i=0}^{N-1} x_i^2 - |x_\lambda| \sum_{i=0}^{N-1} |x_\lambda| \right) & , \text{ si } x_\lambda > 0 \\ 0 & , \text{ si } x_j = 0 \\ -\frac{1}{\|x\|_2^3} \left(\sum_{i=0}^{N-1} x_i^2 - |x_\lambda| \sum_{i=0}^{N-1} |x_\lambda| \right) & , \text{ si } x_\lambda < 0 \end{cases}$$

Ainsi, $\frac{\partial g(x)}{\partial x_\lambda} = 0$ si et seulement si $x_\lambda = 0$ ou si :

$$\sum_{i=0}^{N-1} x_i^2 - |x_\lambda| \sum_{i=0}^{N-1} |x_\lambda| = 0 \quad \forall x_\lambda \in \mathbb{R}^* \Leftrightarrow |x_\lambda| = \frac{\|x\|_2^2}{\|x\|_1}$$

La fonction g , qui n'est pas définie pour $x = \vec{0}$, atteint donc un maximum ou un minimum lorsque les valeurs absolues de tous les éléments de x sont non nuls et égaux entre eux.

Dans ce cas, $\|x\|_1 = N|x_\lambda|$ et $\|x\|_2 = \sqrt{Nx_\lambda^2} = \sqrt{N}|x_\lambda|$ ce qui entraîne :

$$\frac{\|x\|_1}{\|x\|_2} = \frac{N|x_\lambda|}{\sqrt{N}|x_\lambda|} = \frac{N}{\sqrt{N}} = \sqrt{N}$$

Le minimum ou le maximum atteint est donc bien \sqrt{N} . Afin de bien prouver que \sqrt{N} est le maximum atteint, il suffit de calculer une valeur de g en utilisant un $x \neq \vec{0}$ quelconque :

Soit $N = 2 \Rightarrow x \in \mathbb{R}^2$ et

$$x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Ainsi $\|x\|_1 = 1$ et $\|x\|_2 = 1 \Rightarrow \frac{\|x\|_1}{\|x\|_2} = 1$ et $1 < \sqrt{2}$ ce qui entraîne que \sqrt{N} est la valeur maximale atteinte par $g(x)$, $\forall x \in \mathbb{R}^{N*}$

□

Lemme E.1.7 :

$$\frac{\langle \tilde{Q}^T x, Q^T x \rangle}{N\|x\|_2^2} < 2 \quad \forall x \in \mathbb{R}^N \quad (\text{E.8})$$

Preuve :

Prouvons maintenant (E.8) quelles que soient les contraintes de localité, c'est-à-dire pour $\overrightarrow{B(\lambda)} = \overrightarrow{R(\lambda)}$ et $\overrightarrow{B(\lambda)} \neq \overrightarrow{R(\lambda)}$.

Selon (E.6) et en utilisant l'inégalité de Cauchy-Swartz, on peut exprimer la relation suivante :

$$\frac{\langle \tilde{Q}^T x, Q^T x \rangle}{N\|x\|_2^2} = \frac{|\langle \tilde{Q}^T x, Q^T x \rangle|}{N\|x\|_2^2} \leq \frac{\|\tilde{Q}^T x\|_2 \|Q^T x\|_2}{N\|x\|_2^2}$$

avec

$$\begin{aligned} \|Q^T x\|_2 &= \left(\sum_{i=0}^{n-1} (Q^T x)_i^2 \right)^{1/2} = \left(\sum_{i=0}^{n-1} \left(\sum_{j=0}^{N-1} Q_{ij}^T x_j \right)^2 \right)^{1/2} \\ &\leq \left| \sum_{i=0}^{n-1} \sum_{j=0}^{N-1} Q_{ji} |x_j| \right| \leq \left| \sum_{j=0}^{N-1} |x_j| \right| \text{ puisque } \sum_{i=0}^{n-1} Q_{ji} = \|Q\|_\infty = 1 \\ &\leq \sum_{j=0}^{N-1} |x_j| = \|x\|_1 \end{aligned}$$

De la même manière, $\|\tilde{Q}^T x\|_2 \leq \|x\|_1$ puisque $\sum_{i=0}^{n-1} \tilde{Q}_{ji} = \|\tilde{Q}\|_\infty = 1$.

$$\text{Ainsi } \frac{\langle Dx, x \rangle}{N\|x\|_2^2} = \frac{|\langle \tilde{Q}^T x, Q^T x \rangle|}{N\|x\|_2^2} \leq \frac{\|\tilde{Q}^T x\|_2 \|Q^T x\|_2}{N\|x\|_2^2} \leq \frac{1}{N} \frac{\|x\|_1^2}{\|x\|_2^2}$$

$$\text{or selon (E.7), } \frac{\|x\|_1}{\|x\|_2} \leq \sqrt{N} \quad \forall x \in \mathbb{R}^N, \text{ donc } \frac{\langle \tilde{Q}^T x, Q^T x \rangle}{N\|x\|_2^2} \leq 1 < 2$$

□

Ainsi, en regroupant (E.6) et (E.8) nous obtenons $0 < \frac{\langle Dx, x \rangle}{N\|x\|_2^2} < 2$ ce qui prouve que $\varrho(S) < 1$ et donc que le processus converge pour des contraintes de localisation *naturelles*.

□

E.2 Méthode séquentielle appliquée aux courbes : Preuve de la proposition 4.3.8

Dans cette partie est donnée la preuve permettant d'exprimer la dernière erreur connue de G_λ avant sa satisfaction durant l'itération t . Elle concerne la méthode séquentielle appliquée à une courbe.

Preuve 2 : Preuve de la proposition 4.3.8

Prouvons que la proposition (4.33) est vraie pour tout $t > 0$ et pour tout $k < \lambda$. La preuve est faite par récurrence sur λ .

– Cas où $\lambda = 0$: $\overrightarrow{\varepsilon_{0,t}} = \overrightarrow{\varepsilon_{0,-1,t}} = \overrightarrow{e_{0,t-1}}$; (L'expression ne dépend pas de \tilde{C})

– Si l'hypothèse (4.33) est vraie pour $\lambda - 1$, l'est-elle encore pour $\lambda > 0$?

$$\begin{aligned} \overrightarrow{\varepsilon_{\lambda,t}} &= \overrightarrow{\varepsilon_{\lambda,\lambda-1,t}} \quad \text{grâce à (4.32), on obtient :} \\ &= \overrightarrow{\varepsilon_{\lambda,\lambda-2,t}} - \frac{D_{\lambda,\lambda-1}}{D_{\lambda-1,\lambda-1}} \overrightarrow{\varepsilon_{\lambda-1,\lambda-2,t}} \\ &= \overrightarrow{\varepsilon_{\lambda,\lambda-3,t}} - \frac{D_{\lambda,\lambda-2}}{D_{\lambda-2,\lambda-2}} \overrightarrow{\varepsilon_{\lambda-2,\lambda-3,t}} - \frac{D_{\lambda,\lambda-1}}{D_{\lambda-1,\lambda-1}} \overrightarrow{\varepsilon_{\lambda-1,\lambda-2,t}} \\ &\vdots \\ &= \underbrace{\overrightarrow{\varepsilon_{\lambda,-1,t}}}_{\overrightarrow{e_{\lambda,t-1}}} - \sum_{l=1}^{\lambda} \frac{D_{\lambda,\lambda-l}}{D_{\lambda-l,\lambda-l}} \overrightarrow{\varepsilon_{\lambda-l,\lambda-l-1,t}} \end{aligned}$$

De plus, comme $l \geq 1$ on a :

$$\overrightarrow{\varepsilon_{\lambda-l,\lambda-l-1,t}} = \overrightarrow{\varepsilon_{\lambda-l,t}} \stackrel{(4.33)}{=} \overrightarrow{e_{\lambda-l,t-1}} - \sum_{k=0}^{\lambda-l-1} \tilde{C}_{\lambda-l,k} \overrightarrow{e_{k,t-1}}$$

Ce qui entraîne :

$$\begin{aligned}
 \overrightarrow{\varepsilon}_{\lambda,t} &= \overrightarrow{e}_{\lambda,t-1} - \sum_{l=1}^{\lambda} \frac{D_{\lambda,\lambda-l}}{D_{\lambda-l,\lambda-l}} \left(\overrightarrow{e}_{\lambda-l,t-1} - \sum_{k=0}^{\lambda-l-1} \tilde{C}_{\lambda-l,k} \overrightarrow{e}_{k,t-1} \right) \\
 &= \overrightarrow{e}_{\lambda,t-1} - \sum_{l=0}^{\lambda-1} \frac{D_{\lambda,l}}{D_{l,l}} \left(\overrightarrow{e}_{l,t-1} - \sum_{k=0}^{l-1} \tilde{C}_{l,k} \overrightarrow{e}_{k,t-1} \right) \\
 &= \overrightarrow{e}_{\lambda,t-1} - \sum_{k=0}^{\lambda-1} \frac{D_{\lambda,k}}{D_{k,k}} \overrightarrow{e}_{k,t-1} + \sum_{l=0}^{\lambda-1} \sum_{k=0}^{l-1} \frac{D_{\lambda,l}}{D_{l,l}} \tilde{C}_{l,k} \overrightarrow{e}_{k,t-1}
 \end{aligned}$$

Dans la double somme, le cas où $l = 0$ n'a pas de sens puisque k évolue de 0 à $l - 1$, on fait donc commencer l à 1.

$$\begin{aligned}
 \overrightarrow{\varepsilon}_{\lambda,t} &= \overrightarrow{e}_{\lambda,t-1} - \sum_{k=0}^{\lambda-1} \frac{D_{\lambda,k}}{D_{k,k}} \overrightarrow{e}_{k,t-1} + \sum_{l=1}^{\lambda-1} \sum_{k=0}^{l-1} \frac{D_{\lambda,l}}{D_{l,l}} \tilde{C}_{l,k} \overrightarrow{e}_{k,t-1} \\
 &= \overrightarrow{e}_{\lambda,t-1} - \sum_{k=0}^{\lambda-1} \frac{D_{\lambda,k}}{D_{k,k}} \overrightarrow{e}_{k,t-1} + \sum_{k=0}^{\lambda-2} \sum_{l=k+1}^{\lambda-1} \frac{D_{\lambda,l}}{D_{l,l}} \tilde{C}_{l,k} \overrightarrow{e}_{k,t-1} \\
 &= \overrightarrow{e}_{\lambda,t-1} - \sum_{k=0}^{\lambda-1} \frac{D_{\lambda,k}}{D_{k,k}} \overrightarrow{e}_{k,t-1} + \sum_{k=0}^{\lambda-2} \overrightarrow{e}_{k,t-1} \sum_{l=k+1}^{\lambda-1} \frac{D_{\lambda,l}}{D_{l,l}} \tilde{C}_{l,k} \\
 &= \overrightarrow{e}_{\lambda,t-1} - \sum_{k=0}^{\lambda-2} \overrightarrow{e}_{k,t-1} \left(\underbrace{\frac{D_{\lambda,k}}{D_{k,k}} - \sum_{l=k+1}^{\lambda-1} \frac{D_{\lambda,l}}{D_{l,l}} \tilde{C}_{l,k}}_{\tilde{C}_{\lambda,k}} \right) - \underbrace{\frac{D_{\lambda,\lambda-1}}{D_{\lambda-1,\lambda-1}}}_{\tilde{C}_{\lambda,\lambda-1}} \overrightarrow{e}_{\lambda-1,t-1} \\
 &= \overrightarrow{e}_{\lambda,t-1} - \sum_{k=0}^{\lambda-1} \tilde{C}_{\lambda,k} \overrightarrow{e}_{k,t-1}
 \end{aligned}$$

Qui permet de conclure.

□

E.3 Méthode séquentielle appliquée aux courbes : Preuve de la proposition 4.3.9

Dans cette partie est donnée la preuve permettant d'exprimer l'erreur attachée à G_{λ} , après t itérations. Cette erreur est calculée en fonction de l'erreur de cette même contrainte, connue après $t - 1$ itérations. Elle concerne la méthode séquentielle appliquée à une courbe.

Preuve 3 : Preuve de la proposition 4.3.9

Par définition, nous savons que $\overrightarrow{e}_{\lambda,t} = \overrightarrow{\varepsilon}_{\lambda,N-1,t}$. Or grâce à (4.32) il est possible de développer cette expression telle que :

$$\begin{aligned}
 \overrightarrow{e_{\lambda,t}} &= \overrightarrow{\varepsilon_{\lambda,N-1,t}} \\
 &= \overrightarrow{\varepsilon_{\lambda,N-2,t}} - \frac{D_{\lambda,N-1}}{D_{N-1,N-1}} \overrightarrow{\varepsilon_{N-1,N-2,t}} \\
 &= \overrightarrow{\varepsilon_{\lambda,N-3,t}} - \frac{D_{\lambda,N-2}}{D_{N-2,N-2}} \overrightarrow{\varepsilon_{N-2,N-3,t}} - \frac{D_{\lambda,N-1}}{D_{N-1,N-1}} \overrightarrow{\varepsilon_{N-1,N-2,t}} \\
 &\quad \vdots \\
 &= \underbrace{\overrightarrow{\varepsilon_{\lambda,-1,t}}}_{\overrightarrow{e_{\lambda,t-1}}} - \sum_{l=1}^N \frac{D_{\lambda,N-l}}{D_{N-l,N-l}} \overrightarrow{\varepsilon_{N-l,N-l-1,t}}
 \end{aligned} \tag{E.9}$$

Selon (4.34), nous pouvons dire que :

$$\overrightarrow{\varepsilon_{N-l,N-l-1,t}} = \overrightarrow{\varepsilon_{N-l,t}} = \sum_{k=0}^{N-l} C_{N-l,k} \overrightarrow{e_{k,t-1}} \tag{E.10}$$

Ainsi, en injectant (E.10) dans (E.9), on a :

$$\begin{aligned}
 \overrightarrow{e_{\lambda,t}} &= \overrightarrow{e_{\lambda,t-1}} - \sum_{l=1}^N \frac{D_{\lambda,N-l}}{D_{N-l,N-l}} \sum_{k=0}^{N-l} C_{N-l,k} \overrightarrow{e_{k,t-1}} = \overrightarrow{e_{\lambda,t-1}} - \sum_{l=1}^N \sum_{k=0}^{N-l} \frac{D_{\lambda,N-l}}{D_{N-l,N-l}} C_{N-l,k} \overrightarrow{e_{k,t-1}} \\
 &= \overrightarrow{e_{\lambda,t-1}} - \sum_{k=0}^{N-1} \left(\sum_{l=1}^{N-k} \frac{D_{\lambda,N-l}}{D_{N-l,N-l}} C_{N-l,k} \right) \overrightarrow{e_{k,t-1}} = \overrightarrow{e_{\lambda,t-1}} - \sum_{k=0}^{N-1} \left(\sum_{l=k}^{N-1} \frac{D_{\lambda,l}}{D_{l,l}} C_{l,k} \right) \overrightarrow{e_{k,t-1}}
 \end{aligned}$$

Or nous savons que $\sum_{l=k}^{N-1} \frac{D_{\lambda,l}}{D_{l,l}} C_{l,k} = \sum_{l=k}^{N-1} D_{\lambda,l} \frac{1}{D_{l,l}} C_{l,k} = [DF^{-1}C]_{\lambda,k}$ puisque F^{-1} est diagonale. Ainsi, comme C est une matrice triangulaire, on peut écrire :

$$\overrightarrow{e_{\lambda,t}} = \overrightarrow{e_{\lambda,t-1}} - \sum_{k=0}^{N-1} [DF^{-1}C]_{\lambda,k} \overrightarrow{e_{k,t-1}} = \overrightarrow{e_{\lambda,t-1}} - \sum_{k=0}^{N-1} \tilde{S}_{\lambda,k} \overrightarrow{e_{k,t-1}}$$

□

BIBLIOGRAPHIE

- [Alboul and Van Damme, 1995] L. Alboul and R. Van Damme, 1995. Polyhedral metrics in surface reconstruction: tight triangulations. *Memorium*, vol. 1275.
- [Alexe and al., 2004] A. Alexe, V. Gaildrat and L. Barthe, 2004. Interactive modelling from sketches using spherical implicit functions. *AFRIGRAPH'04*, pages 25–34, New York. ACM Press.
- [Amenta and al., 1998] N. Amenta, M. Bern and M. Kamvysselis, 1998. A new voronoi-based surface reconstruction algorithm. *Computer Graphics (ACM SIGGRAPH Proceedings)*, page 415–421.
- [Angelidis and al., 2004] A. Angelidis, G. Wyvill and M.-P. Cani, 2004. Sweepers: swept user-defined tools for modeling by deformation. *ACM Symposium on Solid Modeling Foundations and CAD/CAM applications*, pages 63–73.
- [Astart, 2004] L. Astart, 2004. Représentation de surfaces ayant une structure arborescente par un modèle paramétrique. Thèse de Doctorat, Université de la Méditerranée (Aix-Marseille II).
- [Attali and Boissonnat, 2003] D. Attali and J.-D. Boissonnat, 2003. Complexity of the Delaunay triangulation of points on polyhedral surfaces. *Discrete and Computational Geometry*, vol. 30, n°3, pages 437–452.
- [Aubert and Bechmann, 1997] F. Aubert and D. Bechmann, 1997. Volume-preserving space deformation. *Computers & Graphics*, vol. 21, n°5, pages 625–639.
- [Ault, 1999] H. K. Ault, 1999. Using geometric constraints to capture design intent. *Journal for Geometry and Graphics*, vol. 3, n°1, pages 39–45.
- [Barr, 1981] A. Barr, 1981. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, vol. 1, pages 11–22.
- [Barsky and DeRose, 1990] B. A. Barsky and T. D. DeRose, 1990. Geometric continuity of parametric curves: Construction of geometrically continuous splines. *IEEE Computer Graphics and Applications*, vol. 10, n°1, pages 60–68.
- [Bartels and Beatty, 1989] R. Bartels and J. Beatty, 1989. A technique for the direct manipulation of spline curves. *Graphics Interface*, pages 33–39.
- [Barthe and Kobbelt, 2004] L. Barthe and L. Kobbelt, 2004. Subdivision scheme tuning around extraordinary vertices. *Computer Aided Geometric Design*, vol. 21, n°6, pages 561–583.

- [Bechmann and al., 1997] D. Bechmann, Y. Bertrand and S. They, 1997. Continuous free form deformation. *Computer Networks & ISDN Systems*, vol. 29, n°14, pages 1715–1725.
- [Bechmann and Gerber, 2003] D. Bechmann and D. Gerber, 2003. Arbitrary shaped deformations with DOGME. *The Visual Computer*, vol. 19, n°2–3, pages 175–186.
- [Berge, 1970] C. Berge, 1970. GRAPHEs ET HYPERGRAPHEs. Dunod, Paris.
- [Bezier, 1970] P. Bezier, 1970. EMPLOI DES MACHINES À COMMANDE NUMÉRIQUE. Masson, Paris.
- [Bezier, 1974] P. Bezier, 1974. Local Deformation of NURBS Curves. R. E. Barnhill and R. F. Riesenfeld (eds.), *Computer Aided Geometric Design*. Academic Press, New York.
- [Blanc, 1994] C. Blanc, 1994. Techniques de modélisation et de déformation de surfaces pour la synthèse d’images. Thèse de Doctorat, Université Bordeaux.
- [Bloomenthal and al., 1997] J. Bloomenthal, C. Bajaj, J. Blinn, M.-P. Cani-Gascuel, A. Rockwood, B. Wyvill and G. Wyvill, 1997. INTRODUCTION TO IMPLICIT SURFACES. Morgan Kaufmann.
- [Bloor and Wilson, 1990] M. I. G. Bloor and M. J. Wilson, 1990. Representing PDE surfaces in terms of B-splines. *Computer-Aided Design*, vol. 22, n°6, pages 324–331.
- [Boehm and Prautzsch, 1985] W. Boehm and H. Prautzsch, 1985. The insertion algorithm. *Computer-Aided Design*, vol. 17, n°2, pages 58–59.
- [Boissonnat et Yvinec, 1995] J.-D. Boissonnat et M. Yvinec, 1995. GÉOMÉTRIE ALGORITHMIQUE. Ediscience International.
- [Bonnefoi and al., 2004] P.-F. Bonnefoi, D. Plemenos and W. Ruchaud, 2004. Declarative modelling in computer graphics: current results and future issues. *International Conference on Computational Science*, pages 80–89.
- [Borges and Pastva, 2002] C. F. Borges and T. Pastva, 2002. Total least squares fitting of Bézier and B-spline curves to ordered data. *Computer Aided Geometric Design*, vol. 19, n°4, pages 275–289.
- [Borrel and Bechmann, 1991] P. Borrel and D. Bechmann, 1991. Deformation of n-dimensional objects. *Shape Modeling and Applications*, pages 351–369, New York. ACM Press.
- [Borrel and Rappoport, 1994] P. Borrel and A. Rappoport, 1994. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics*, vol. 2, n°13, pages 137–155.
- [Botsch and Kobbelt, 2004] M. Botsch and L. Kobbelt, 2004. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics*, vol. 23, n°3, pages 630–634.
- [Boughanem et Plemenos, 1994] M. Boughanem et D. Plemenos, 1994. Techniques d’apprentissage en modélisation déclarative de scènes par décomposition hiérarchique. *International Conference on Computer Graphics and Artificial Intelligence*, pages 99–120.

- [Bouvet, 1988] G. Bouvet, 1988. Techniques de modélisation déclarative d'ensembles de plaquettes polygonales régulières. Rapport de DEA, Université de Rennes.
- [Braemer et Kerbrat, 1976] J.-M. Braemer et Y. Kerbrat, 1976. GÉOMÉTRIE DES COURBES ET DES SURFACES. Hermann, Paris.
- [Cani, 1999] M.-P. Cani, 1999. Implicit representations in computer animation: a compared study. *Implicit Surface*. Invited paper.
- [Catmull and Clark, 1978] E. Catmull and J. Clark, 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, vol. 10, n°6, pages 350–355.
- [Caubet et al., 1992] R. Caubet, N. E. Djedi, V. Gaildrat, F. Rubio, G. Pérennou et N. Vigouroux, 1992. NOVAC : Un outil d'aide à la création graphique pour non-voyants. *Interface to Real & Virtual worlds*, Montpellier.
- [Chaikin, 1974] G. Chaikin, 1974. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, vol. 3, pages 346–349.
- [Chambelland, 2001] J.-C. Chambelland, 2001. Approche déclarative de la modélisation de surfaces. Rapport de DEA, Université de la Méditerranée (Aix-Marseille II).
- [Champciaux, 1997] L. Champciaux, 1997. Declarative modelling: speeding up the generation. *International Conference on Imaging Science, Systems and Technology*.
- [Champciaux, 1998] L. Champciaux, 1998. Introduction de techniques d'apprentissage en modélisation déclarative. Thèse de Doctorat, École des Mines de Nantes.
- [Chauvat, 1994] D. Chauvat, 1994. Le projet VoluFormes : un exemple de modélisation déclarative avec contrôle spatial. Thèse de Doctorat, Université de Nantes.
- [Cheutet et al., 2004] V. Cheutet, J.-P. Pernot, J.-C. Léon, B. Falcidieno and F. Giannini, 2004. Inserting G1 discontinuities into free-form features for aesthetic design. *WSCG'04*, pages 109–116.
- [Chomsky, 1956] N. Chomsky, 1956. Three models for the description of language. *IRA Transactions on Information Theory*, vol. 2, n°3, pages 113–124.
- [Christianson et al., 1996] D. B. Christianson, S. E. Anderson, L.-W. He, D. S. Weld, M. F. Cohen and D. H. Salesin, 1996. Declarative camera control for automatic cinematography. *American Association for Artificial Intelligence*, pages 148–155.
- [Ciarlet, 1998] P. G. Ciarlet, 1998. INTRODUCTION À L'ANALYSE NUMÉRIQUE MATRI-CIELLE ET À L'OPTIMISATION. Dunod, Paris.
- [Clouchoux et al., 2004] C. Clouchoux, O. Coulon, A. Cachia, D. Riviere, J.-F. Mangin and J. Regis, 2004. Towards an anatomically meaningful parameterization of the cortical surface. *Medical Image Computing and Computer-Assisted Intervention*, vol. 3217 / 2004, pages 1046–1047. Springer-Verlag.

- [Cobb, 1984] E. S. Cobb, 1984. Design of sculptured surfaces using the B-spline representation. PhD, University of Utah.
- [Cocquerez et Philipp, 1998] J.-P. Cocquerez et S. Philipp, 1998. ANALYSE D'IMAGES : FILTRAGE ET SEGMENTATION. Masson, Paris.
- [Cohen and Merrien, 2003] A. Cohen and J.-L. Merrien (eds.), 2003. CURVE AND SURFACE FITTING: SAINT-MALO 2002. Nashboro Press.
- [Colin, 1990a] C. Colin, 1990. Calcul automatique d'une bonne vue d'une scène. *MICAD*.
- [Colin, 1990b] C. Colin, 1990. Modélisation déclarative de scènes à base de polyèdres élémentaires. Thèse de Doctorat, Université de Rennes.
- [Colin and al., 1997] C. Colin, E. Desmontils, J. Martin and J. Mounier, 1997. Working modes with a declarative modeler. *Compugraphics*.
- [Coquillart, 1990] S. Coquillart, 1990. Extended free-form deformation: a sculpturing tool for 3D geometric modeling. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 24, n°4, pages 187–196.
- [Crespin, 1998] B. Crespin, 1998. Modélisation et déformation de forme libre à base de surfaces implicites équipotentielles. Thèse de Doctorat, Université Bordeaux.
- [Cruz-Neira and al., 1993] C. Cruz-Neira, D. Sandin and T. DeFanti, 1993. Virtual Reality : the design and implementation of the CAVE. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 27, n°2, pages 135–142.
- [Cutler and al., 1997] D. L. Cutler, B. Fröhlich and P. Hanrahan, 1997. Two-handed direct manipulation on the responsive workbench. *Symposium on Interactive 3D graphics*.
- [Daniel, 1989] M. Daniel, 1989. Modélisation de courbes et surfaces par B-splines. Application à la conception et à la visualisation de formes. Thèse de Doctorat, Université de Nantes, École Nationale Supérieure de Mécanique.
- [Daniel, 1996] M. Daniel, 1996. Declarative Modeling of fair shapes: An additional approach to curves and surfaces computations. J. Hoschek and P. Kaklis (eds.), *Advanced Course on FAIRSHAPE*, pages 77–85. B. G. Teuber Stuttgart.
- [De Boor, 1978] C. De Boor, 1978. A PRACTICAL GUIDE TO SPLINES. Springer-Verlag.
- [De Mori, 1998] R. De Mori, 1998. SPOKEN DIALOGUES WITH COMPUTERS. Academic Press.
- [Debunne, 2000] G. Debunne, 2000. Animation multirésolution d'objets déformables en temps-réel, application à la simulation chirurgicale. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- [Delobel, 2000] F. Delobel, 2000. Résolution de systèmes de contraintes réelles non linéaires. Thèse de Doctorat, Université de Nice Sophia Antipolis.
- [Demengel, 1998] G. Demengel, 1998. MODÈLES DE BÉZIER, DES B-SPLINES ET DES NURBS: OUTILS POUR L'INGÉNIEUR, BASES POUR LA CAO : MATHÉMATIQUES DES COURBES ET DES SURFACES. Ellipses, Paris.

- [Demoisson, 1994] C. Demoisson, 1994. Vers la modélisation déclarative de courbes. Rapport de DEA, Université de Nantes, École Centrale de Nantes.
- [Denand, 2004] N. Denand, 2004. Traitement automatique de phrases locatives statiques du français. Thèse de Doctorat, Université de la Méditerranée (Aix-Marseille II).
- [Desmontils, 1995a] E. Desmontils, 1995. Formalisation des propriétés en modélisation déclarative à l'aide des ensembles flous. Rapport de Recherche IRIN-106, École Centrale de Nantes, Université de Nantes.
- [Desmontils, 1995b] E. Desmontils, 1995. Les modeleurs déclaratifs. Rapport de Recherche IRIN-95, École Centrale de Nantes, Université de Nantes.
- [Desmontils, 1996] E. Desmontils, 1996. Une formalisation des propriétés en modélisation déclarative à l'aide des ensembles flous. *International Conference on Computer Graphics and Artificial Intelligence*.
- [Desmontils, 1998] E. Desmontils, 1998. Le projet CordiFormes : une plate-forme pour la construction de modeleurs déclaratifs. Thèse de Doctorat, Université de Nantes.
- [Desmontils, 2000] E. Desmontils, 2000. Expressing constraint satisfaction problems in declarative modeling using natural language and fuzzy sets. *Computers & Graphics*, vol. 24, n°4, pages 555–568.
- [Desmontils et Martin, 1997] E. Desmontils et J.-Y. Martin, 1997. La propriété dans tous ses états. Rapport de Recherche IRIN-148, IRIN - MGII, Nantes.
- [Desmontils et Pacholczyk, 1997a] E. Desmontils et D. Pacholczyk, 1997. Interprétation du vague dans la description linguistique d'une scène en Modélisation Déclarative. Rapport de Recherche IRIN-150, Université de Nantes.
- [Desmontils et Pacholczyk, 1997b] E. Desmontils et D. Pacholczyk, 1997. Vers un traitement linguistique des propriétés en modélisation déclarative. *Revue Internationale de CFAO et d'Informatique Graphique*, vol. 12, n°4, pages 351–371.
- [Djedi, 1991] N. Djedi, 1991. Modélisation en synthèse d'images : utilisation d'une méthodologie déclarative. Thèse de Doctorat, Université Paul Sabatier (Toulouse III).
- [Do Carmo, 1976] Do Carmo, 1976. DIFFERENTIAL GEOMETRY OF CURVES AND SURFACES. Prentice Hall.
- [Doo and Sabin, 1978] D. Doo and M. Sabin, 1978. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, vol. 10, n°6, pages 356–360.
- [Doss-Bachelet et al., 2000] C. Doss-Bachelet, J.-P. Françoise et C. Piquet, 2000. GÉOMÉTRIE DIFFÉRENTIELLE. Ellipses, Paris.
- [Dragicevic, 2004] P. Dragicevic, 2004. Un modèle d'interaction en entrée pour des systèmes interactifs multi-dispositifs hautement configurables. Thèse de Doctorat, Université de Nantes.

- [Dyllong and Luther, 1999] E. Dyllong and W. Luther, 1999. Distance Calculation Between a Point and a NURBS Surface. P.-J. Laurent, P. Sablonniere, L. L. Schumaker and P. Sablonniere (eds.), *Curve and Surface Design: Saint-Malo 1999*, pages 55–62. Vanderbilt University Press.
- [Dyn and al., 1996] N. Dyn, J. Gregory and D. Levin, 1996. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, vol. 9, pages 160–169.
- [Edwards et al., 1995] G. Edwards, G. Ligozat, A. Gryl, L. Fraczak, B. Moulin et C. M. Gold, 1995. Un modèle pivot pour la représentation de l'espace basé sur les diagrammes de Voronoï : application aux descriptions d'itinéraires en langage naturel. *Journées Cassini*.
- [Estratrat, 2003] M. Estratrat, 2003. Application de la configuration à l'analyse syntaxico-sémantique de Description. Rapport de DEA, Universités de Provence (Aix-Marseille I), de la Méditerranée (Aix-Marseille II), Paul Cézanne (Aix-Marseille III), ENSAM.
- [Estratrat et Henocque, 2005] M. Estratrat et L. Henocque, 2005. Comprendre des descriptions simples à l'aide d'un configurateur. *Journées Francophones de Programmation par Contraintes*, pages 325–334, Université d'Artois.
- [Farin, 2002] G. Farin, 2002. CURVES AND SURFACES FOR CAGD - A PRACTICAL GUIDE - FIFTH EDITION. Morgan Kaufmann.
- [Feng and al., 1996] J. Feng, L. Ma and Q. Peng, 1996. A new free-form deformation through the control of parametric surfaces. *Computers & Graphics*, vol. 20, n°4.
- [Ferley and al., 1997] E. Ferley, M.-P. Cani-Gascuel and D. Attali, 1997. Skeletal reconstruction of branching shapes. *Computer Graphics Forum*, vol. 16, n°5, pages 283–293.
- [Foley and al., 1997] J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes, 1997. COMPUTER GRAPHICS - PRINCIPLES AND PRACTICE. Addison-Wesley, Boston.
- [Fowler, 1992] B. Fowler, 1992. Geometric manipulation of tensor product surfaces. *Symposium on Interactive 3D graphics*, pages 101–108. ACM Press.
- [Fowler and Bartels, 1993] B. Fowler and R. Bartels, 1993. Constraint based curve manipulation. *IEEE Computer Graphics and Applications*, vol. 13, n°5, pages 43–49.
- [Fraczak, 1994] L. Fraczak, 1994. De la description au croquis. Rapport de DEA, Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur.
- [Gaildrat, 2003] V. Gaildrat, 2003. Modélisation déclarative d'environnements virtuels : Création de scènes et de formes complexes par l'énoncé de propriétés et l'emploi d'interactions gestuelles. Habilitation à Diriger des Recherches, Université Paul Sabatier (Toulouse III).
- [Gaildrat et al., 1993] V. Gaildrat, N. Vigouroux, R. Caubet et G. Perennou, 1993. Conception d'une interface multimodale pour un modeleur déclaratif de scènes tridimensionnelles pour la synthèse d'images. *Interface to Real & Virtual worlds*, Montpellier.

- [Garland and Heckbert, 1997] M. Garland and P. Heckbert, 1997. Surface simplification using quadric error metrics. *Computer Graphics (ACM SIGGRAPH Proceedings)*, page 209–216.
- [Gesquiere, 2000] G. Gesquiere, 2000. Calcul et contrôle de volume d’objets implicites. Thèse de Doctorat, Université de Dijon.
- [Goldenthal and Bercovier, 2004] R. Goldenthal and M. Bercovier, April 2004. Spline curve approximation and design over the knots computing. *The Visual Computer*, vol. 72, n°1-2, pages 53–64.
- [Gondran et Minoux, 1979] M. Gondran et M. Minoux, 1979. GRAPHERS ET ALGORITHMES. Eyrolles.
- [Grandcolas and al., 2003] S. Grandcolas, L. Henocque and N. Prcovic, 2003. A Canonicity test for configuration. *International Conference on Principles and Practice of Constraint Programming*, pages 853–857.
- [Gryl and Gonçalves, 1995] A. Gryl and M.-R. Gonçalves, 1995. Spatial representations and route descriptions. *International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, pages 813–818.
- [Guillet, 1999] S. Guillet, 1999. Modification et construction de formes gauches soumises à des contraintes de conception. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- [Guinier, 1992] D. Guinier, 1992. SÉCURITÉ ET QUALITÉ DES SYSTÈMES D’INFORMATION - APPROCHE SYSTÈME. Masson.
- [Hahmann et al., 2005] S. Hahmann, G.-P. Bonneau et B. Sauvage, 2005. Déformation de courbes multirésolution sous contraintes. *Revue Internationale d’Ingénierie Numérique*, vol. 1, n°1, pages 77–90.
- [Hahmann and Elber, 2004] S. Hahmann and G. Elber, 2004. Constrained Multiresolution Geometric Modeling. N. A. Dodson, M. A. Sabin and M. Floater (eds.), *Advances in Multiresolution for Geometric Modeling*. Springer-Verlag.
- [Hahmann and al., 2005] S. Hahmann, B. Sauvage and G.-P. Bonneau, 2005. Area preserving deformation of multiresolution curves. *Computer Aided Geometric Design*, vol. 22, pages 349–367.
- [Hanson and Lawson, 1969] R. J. Hanson and C. L. Lawson, 1969. Extensions and applications of the Householder algorithm for solving linear least squares problems. *Mathematics of Computation*, vol. 23, n°108, pages 787–812.
- [Henocque et al., 2005] L. Henocque, M. Kleiner et N. Prcovic, 2005. Une procédure générale d’élimination d’isomorphismes pour les problèmes de configuration. *Journées Francophones de Programmation par Contraintes*, pages 335–344.
- [Hoppe, 1994] H. Hoppe, 1994. Surface reconstruction from unorganized points. PhD, University of Washington.

-
- [Hormann, 2001] K. Hormann, 2001. Theory and applications of parameterizing triangulations. PhD, University of Erlangen.
- [Hornus and al., 2003] S. Hornus, A. Angelidis and M.-P. Cani, 2003. Implicit modelling using subdivision-curves. *The Visual Computer*, vol. 19, n°2-3, pages 94–104.
- [Hsu and al., 1992] W. M. Hsu, J. F. Hughes and H. Kaufman, 1992. Direct manipulation of free-form deformations. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 26, n°2, pages 177–184.
- [Huot and al., 2004] S. Huot, C. Dumas, P. Dragicevic, J.-D. Fekete and G. Hégron, 2004. The MaggLite Post-WIMP Toolkit: draw It, connect It and run It. *Computer Graphics (ACM SIGGRAPH Proceedings)*, pages 257–266.
- [Huot et al., 2004] S. Huot, C. Dumas et G. Hégron, 2004. Svalabard : une table à dessin virtuelle pour la modélisation 3D. *Interaction Homme-Machine*, pages 85–92, Namur.
- [Jermann, 2002] C. Jermann, 2002. Résolution de contraintes géométriques par rigidification récursive et propagation d’intervalles. Thèse de Doctorat, Université de Nice Sophia Antipolis.
- [Jolivet and al., 2004] V. Jolivet, D. Plemenos and P. Poulingeas, 2004. Declarative specification of ambiance in VRML landscapes. *International Conference on Computational Science*, pages 115–122.
- [Kazinnik and Elber, 1997] R. Kazinnik and G. Elber, 1997. Orthogonal decomposition of non-uniform B-spline spaces using wavelets. *Computer Graphics Forum*, vol. 16, n°3, pages 27–38.
- [Kobbelt, 2000] L. Kobbelt, 2000. $\sqrt{3}$ -subdivision. *Computer Methods in Applied Mechanics and Engineering*.
- [Kobbelt and al., 2000] L. Kobbelt, S. Bischoff, M. Botsch, K. Kähler, C. Rössl and Robert, 2000. Geometric modeling based on polygonal meshes. *Eurographics Tutorial*.
- [Kwaiter, 1998] G. Kwaiter, 1998. Modélisation déclarative de scènes : étude et réalisation de solveurs de contraintes. Thèse de Doctorat, Université Paul Sabatier (Toulouse III).
- [Kwaiter and al., 1997] G. Kwaiter, V. Gaildrat and R. Caubet, 1997. Dynamic and hierarchical constraints solver with continuous variables. *French Conference on Logic Programming and Constraint Programming*.
- [La Greca, 2001] R. La Greca, 2001. Segmentation de nuages de points et courbures discrètes pour la reconstruction. Rapport de DEA, Université de la Méditerranée (Aix-Marseille II).
- [La Greca et Daniel, 2002] R. La Greca et M. Daniel, 2002. Modélisation déclarative : de la description vers les modèles. *Journées des Doctorants du LSIS*, pages 3–12.
- [La Greca and Daniel, 2004a] R. La Greca and M. Daniel, 2004. Declarative approach to NURBS surface design: from semantic to geometric models. *International Conference on Computer Graphics and Artificial Intelligence*, pages 161–168, Limoges.

- [La Greca et Daniel, 2004] R. La Greca et M. Daniel, 2004. Déformation de courbes NURBS sous contraintes. *Journées du Groupe de Travail Modélisation Géométrique GDR ALP*, pages 162–171, Lyon.
- [La Greca and Daniel, 2004b] R. La Greca and M. Daniel, 2004. Local deformation of NURBS curves to satisfy position constraints. *Mathematical Methods for Curves and Surfaces*, Tromso.
- [La Greca and al., 2004] R. La Greca, M. Daniel and A. Bac, 2004. Satisfying constraints through local deformation of NURBS curves. Technical Report LSIS.RR.2004.006, Laboratoire des Sciences de l'Information et des Systèmes.
- [La Greca and al., 2005] R. La Greca, M. Daniel and A. Bac, 2005. Local Deformation of NURBS Curves. M. Daehlen, K. Morken and L. L. Schumaker (eds.), *Mathematical Methods for Curves and Surfaces: Tromso 2004*, pages 243–252. Nashboro Press, Brentwood, TN.
- [La Greca et al., 2002] R. La Greca, M. Daniel et P. Véron, 2002. La description dans l'approche déclarative de la modélisation de surfaces. *Journées du Groupe de Travail Modélisation Géométrique GDR ALP*, pages 46–55, Nantes.
- [Labelle, 1981] J. Labelle, 1981. THÉORIE DES GRAPHS. Modulo, Québec.
- [Lauriere, 1976] J. L. Lauriere, 1976. Un langage et un programme pour résoudre et énoncer des problèmes combinatoires : ALICE. Thèse d'État, Université Paris 6.
- [Lazarus and al., 1994] F. Lazarus, S. Coquillart and P. Jancene, 1994. Axial deformation: An intuitive deformation technique. *Computer-Aided Design*, vol. 26, n°8, pages 607–613.
- [Le Goff, 1990] D. Le Goff, 1990. Modélisation déclarative et morphologie urbaine. Rapport de DEA, Université de Nantes.
- [Le Roux, 2003] O. Le Roux, 2003. Modélisation déclarative d'environnements virtuels : contribution à l'étude des techniques de génération par contraintes. Thèse de Doctorat, Université Paul Sabatier (Toulouse III).
- [Le Roux and al., 2004] O. Le Roux, V. Gaildrat and R. Caubet, 2004. Using Constraint Satisfaction Techniques in Declarative Modelling. D. Hardbound (ed.), *Geometric Modeling: Techniques, Applications, Systems and Tools*. Kluwer Academic.
- [Léon, 1991] J.-C. Léon, 1991. MODÉLISATION ET CONSTRUCTION DE SURFACES POUR LA CFAO. Hermes.
- [Lesage, 2002] D. Lesage, 2002. Un modèle dynamique de spécifications d'ingénierie basé sur une approche de géométrie variationnelle. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- [Leyton, 1988] M. Leyton, 1988. A process-grammar for shape. *Artificial Intelligence*, vol. 34, pages 213–247.
- [Leyton, 1992] M. Leyton, 1992. SYMMETRY, CAUSALITY, MIND. The MIT Press.

- [Leyton, 2001] M. Leyton, 2001. A GENERATIVE THEORY OF SHAPE. Springer.
- [Lhomme et Rueher, 1997] O. Lhomme et M. Rueher, 1997. Application des techniques CSP au raisonnement sur les intervalles. *Revue d'Intelligence Artificielle*, vol. 11, n°3, pages 283–311.
- [Ligozat, 1994] G. Ligozat, 1994. REPRÉSENTATION DES CONNAISSANCES LINGUISTIQUES. Armand Colin, Paris.
- [Littré, 1872] É. Littré, 1872. DICTIONNAIRE LE LITTRÉ.
- [Loop, 1987] C. T. Loop, 1987. Smooth subdivision surfaces based on triangles. Master of Science, University of Utah.
- [Lorensen and Cline, 1987] W. E. Lorensen and H. E. Cline, 1987. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 21, n°4, pages 163–169.
- [Lounsbery and al., 1997] M. Lounsbery, T. D. DeRose and J. Warren, 1997. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, vol. 16, n°1, pages 34–73.
- [Lucas, 1993] M. Lucas, 1993. Conception assistée par ordinateur et modélisation déclarative de formes. *Colloque du Pôle de Ressources Informatiques pour la MECA-nique*, pages 93 – 98, Chatenay-Malabry.
- [Lucas et Desmontils, 1995] M. Lucas et E. Desmontils, 1995. Les modeleurs déclaratifs. *Revue Internationale de CFAO et d'Informatique Graphique*, vol. 10, n°6, pages 559–586.
- [Lucas et al., 1990] M. Lucas, D. Martin, P. Martin et D. Plemenos, 1990. Le projet Ex-ploFormes : quelques pas vers la modélisation déclarative de formes. *Journées AFCET-GROPLAN, Strasbourg, 1989, publié dans BIGRE N ° 67*, pages 35–49.
- [Lucas et Poulet, 1994] M. Lucas et F. Poulet, 1994. Modélisation déclarative de monuments mégalithiques : le projet MégaFormes. Rapport de Recherche IRIN - 58, École Centrale de Nantes.
- [Léon and Trompette, 1995] J.-C. Léon and P. Trompette, 1995. A new approach towards free-form surfaces control. *Computer Aided Geometric Design*, vol. 12, pages 395–416.
- [Maculet, 1991] R. Maculet, 1991. IA et CAO en architecture : représentation des connaissances spatiales et raisonnement spatial avec contraintes. Thèse de Doctorat, Université Paris 6.
- [Maculet, 1992] R. Maculet, 1992. Algèbre de Manhattan, satisfaction de contraintes et placements d'objets sous contraintes. *GROPLAN*.
- [Malik et al., 2002] E. Malik, Y. Gardan et E. Perrin, 2002. Vers une approche plus intuitive des systèmes de CAO. *Journées de l'Association Française d'Informatique Graphique*, pages 79–87.

- [Mallat, 1989] S. Mallat, 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pages 674–693.
- [Maltret et Daniel, 2001] J.-L. Maltret et M. Daniel, 2001. Courbures discrètes et applications. *Journées du Groupe de Travail Modélisation Géométrique GDR ALP*, Dijon. 17 pages.
- [Maltret and Daniel, 2002a] J.-L. Maltret and M. Daniel, 2002. Discrete curvatures and applications: a survey. Technical Report LSIS.RR.2002.002, Laboratoire des Sciences de l'Information et des Systèmes.
- [Maltret and Daniel, 2002b] J.-L. Maltret and M. Daniel, 2002. Local analysis of polyhedral surfaces. *Fifth International Conference "Curves and Surfaces"*, Saint Malo.
- [Marciniak and Vetulani, 2002] J. Marciniak and Z. Vetulani, 2002. Ontology of spatial concepts in a natural language interface for a mobile Robot. *Applied Intelligence*, vol. 17, n°3, pages 271–274.
- [Mari, 2002] J.-L. Mari, 2002. Modélisation de formes complexes intégrant leurs caractéristiques globales et leurs spécificités locales. Thèse de Doctorat, Université de la Méditerranée (Aix-Marseille II).
- [Martin, 1990] J.-Y. Martin, 1990. Synthèse d'images à l'aide d'automates cellulaires. Thèse de Doctorat, Université de Rennes.
- [Martin and Martin, 1993] P. Martin and D. Martin, 1993. Declarative generation of a family of polyhedra. *GRAPHICON*, Saint-Petersbourg.
- [Martin and Martin, 1999] P. Martin and D. Martin, 1999. PolyFormes: software for the declarative modelling of polyhedra. *The Visual Computer*, vol. 15, n°2, pages 55–76.
- [Marvie and al., 2005] J.-E. Marvie, J. Perret and K. Bouatouch, 2005. The FL-system: a functional L-system for procedural geometric modeling. *The Visual Computer*, vol. 21, n°5, pages 329–339.
- [Maya, 2002] V. Maya, 2002. Processing information of geographical databases within a dialogue system: a pivot system. *European Conference on Artificial Intelligence*, pages 475–479.
- [Moron, 1996] V. Moron, 1996. Mise en correspondance de données 3D avec un modèle CAO. Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon.
- [Naur and al., 1963] P. Naur, J. W. Backus, J. H. Wegstein, A. van Wijngaarden, M. Woodger and al., 1963. Revised report on the algorithmic language ALGOL 60. *Communications of the ACM*, vol. 6(1), pages 1–17.
- [Nussbaumer, 1982] H. J. Nussbaumer, 1982. FAST FOURIER TRANSFORMS AND CONVOLUTION ALGORITHMS. Springer, New York.
- [Pace, 1996] A. Pace, 1996. Modélisation déclarative et contraintes fonctionnelles. Rapport de DEA, École Centrale de Nantes.

- [Pajot-Duval, 1994] L. Pajot-Duval, 1994. Modélisation déclarative de configuration de segments de droite. Thèse de Doctorat, Université de Nantes.
- [Parent, 1977] R. Parent, 1977. A system for sculpting 3d data. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 11, n°2, pages 138–147.
- [Parisot et al., 2002] C. Parisot, V. Verdoux, R. La Greca, P. Véron et M. Daniel, 2002. Description de formes en mécanique. Rapport de Recherche LSIS.RR.2002.008, Laboratoire des Sciences de l'Information et des Systèmes.
- [Pasko and al., 1995] A. Pasko, V. Adzhiev and V. Savchenko, 1995. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer*, vol. 11, n°8, pages 429–446.
- [Pernot, 2004] J.-P. Pernot, 2004. Fully Free Form Deformation Features for Aesthetic and Engineering Designs. PhD, Institut National Polytechnique de Grenoble.
- [Pernot and al., 2003] J.-P. Pernot, S. Guillet, J.-C. Léon, B. Falcidieno and F. Giannini, 2003. A new approach to minimisations for shape control during free-form surface deformation. *International 2003 Design Engineering Technical Conferences And Design and Automation Conference*.
- [Petit et Morteau, 1996] S. Petit et X. Morteau, 1996. Description de courbes. Rapport de stage de l'École des Beaux-Arts, École Centrale de Nantes.
- [Piegl and Tiller, 1997] L. Piegl and W. Tiller, 1997. THE NURBS BOOK 2ND EDITION. Springer, Berlin.
- [Planche, 1988] R. Planche, 1988. MAÎTRISER LA MODÉLISATION CONCEPTUELLE. Masson.
- [Postnikov, 1990] M. Postnikov, 1990. LEÇONS DE GÉOMÉTRIE - VARIÉTÉS DIFFÉRENTIABLES. Mir, Moscou.
- [Poulet, 1994] F. Poulet, 1994. Modélisation déclarative de scènes tridimensionnelles par énumération spatiale : le projet SpatioFormes. Thèse de Doctorat, Université de Rennes.
- [Prusinkiewicz and al., 1999] P. Prusinkiewicz, J. Hanan and R. Mech, 1999. An L-system-based plant modeling language. *Applications of Graph Transformations with Industrial Relevance*, vol. Lecture Notes in Computer Science 1779, pages 395–410.
- [Prusinkiewicz and Lindenmayer, 1990] P. Prusinkiewicz and A. Lindenmayer, 1990. THE ALGORITHMIC BEAUTY OF PLANTS. Springer-Verlag, New York.
- [Qin and Terzopoulos, 1996] H. Qin and D. Terzopoulos, 1996. D-NURBS: a physics-based framework for geometric design. *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, n°1, pages 85–96.
- [Raffin, 2000] R. Raffin, 2000. Déformation sous contraintes généralisées. Thèse de Doctorat, Université de Dijon.

- [Randrianarivony and Brunnett, 2002] M. Randrianarivony and G. Brunnett, 2002. Parallel implementation of surface reconstruction from noisy samples. *SFB Preprint*, pages 02–16.
- [Robillard, 1996] E. Robillard, 1996. Modélisation déclarative de courbes. Rapport de DEA, Université de Nantes, École Centrale de Nantes.
- [Rochfeld et Moréjon, 1986] A. Rochfeld et J. Moréjon, 1986. LA MÉTHODE MERISE TOME 3, GAMME OPÉRATOIRE. Les Éditions d'Organisation.
- [Rogers, 2000] D. F. Rogers, 2000. AN INTRODUCTION TO NURBS: WITH HISTORICAL PERSPECTIVE - 1ST EDITION. Morgan Kaufmann.
- [Rossignol, 2001] V. Rossignol, 2001. Approche déclarative de la modélisation de courbes. Thèse de Doctorat, Université de Nantes.
- [Ruchaud, 2001] W. Ruchaud, 2001. Étude et réalisation d'un moteur de résolution de contraintes géométriques pour la modélisation déclarative. Thèse de Doctorat, Université de Limoges.
- [Ruchaud and Plemenos, 2002] W. Ruchaud and D. Plemenos, 2002. MultiFormes: a declarative modeller as a 3D scene sketching tool. *International Conference on Computer Vision and Graphics*.
- [Saporta, 1996] G. Saporta, 1996. PROBABILITÉS, ANALYSE DES DONNÉES ET STATISTIQUE. Technip, Paris.
- [Schek, 1974] H. J. Schek, 1974. The force density method for form-finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering*, vol. 3, n°2, pages 115–134.
- [Schichtel, 1993a] M. Schichtel, 1993. Lifting and adjusting of planar freeform curves: a way to modify curves preserving their shape. Technical report, Darmstadt University of Technology.
- [Schichtel, 1993b] M. Schichtel, 1993. Lifting of freeform-surfaces. Technical report, Darmstadt University of Technology.
- [Schroeder and al., 1992] W. Schroeder, J. Zarge and W. Lorensen, 1992. Decimation of triangle meshes. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 26, n°2, pages 65–70.
- [Sclaroff and Pentland, 1991] S. Sclaroff and A. Pentland, 1991. Generalized implicit functions for computer graphics. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 25, n°4, pages 247–250.
- [Sederberg and Parry, 1986] T. W. Sederberg and S. R. Parry, 1986. Free-form deformation of solid geometric models. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 20, n°4, pages 151–160.
- [Sherstyuk, 1999] A. Sherstyuk, 1999. Fast ray tracing of implicit surfaces. *Computer Graphics Forum*, vol. 18, n°2, pages 139–147(9).

- [Spriet and El-Beze, 1998] T. Spriet and M. El-Beze, 1998. Introduction of Rules into a Stochastic Approach for Language Modelling. K. Ponting (ed.), *Computational Models of Speech Pattern Processing*, vol. 169, pages 350–355. NATO ASI Series F.
- [Stam, 1998] J. Stam, 1998. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. *Computer Graphics (ACM SIGGRAPH Proceedings)*, pages 395–404.
- [Stollnitz and al., 1996] E. Stollnitz, T. DeRose and D. H. Salesin, 1996. WAVELETS FOR COMPUTER GRAPHICS: THEORY AND APPLICATIONS. Morgan Kaufmann, San Francisco.
- [Tardieu et al., 2001] H. Tardieu, D. Nanci, B. Espinasse et J.-L. Le Moigne, 2001. INGÉNIERIE DES SYSTÈMES D'INFORMATION : MERISE. Vuibert.
- [Tardieu et al., 1983] H. Tardieu, A. Rochfeld, R. Colletti, G. Panet et G. Vahée, 1983. LA MÉTHODE MERISE TOME 1, PRINCIPES ET OUTILS. Les Éditions d'Organisation.
- [Tardieu et al., 1985] H. Tardieu, A. Rochfeld, R. Colletti, G. Panet et G. Vahée, 1985. LA MÉTHODE MERISE TOME 2, DÉMARCHE ET PRATIQUES. Les Éditions d'Organisation.
- [Terzopoulos and al., 1987] D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, 1987. Elastically deformable models. *Computer Graphics (ACM SIGGRAPH Proceedings)*, vol. 21, n°4, pages 205–214.
- [Terzopoulos and Qin, 1994] D. Terzopoulos and H. Qin, 1994. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, vol. 13, n°2, pages 103–136.
- [Thingvold and Cohen, 1990] J. A. Thingvold and E. Cohen, 1990. Physical modeling with B-spline surfaces for interactive design and animation. *Symposium on Interactive 3D graphics*, pages 129–137, New York. ACM Press.
- [Tollenaere et al., 1998] M. Tollenaere et al., 1998. CONCEPTION DE PRODUITS MÉCANIQUES, MÉTHODES, MODÈLES ET OUTILS. Hermès, Paris.
- [Valiron, 1945] G. Valiron, 1945. ÉQUATIONS FONCTIONNELLES - APPLICATIONS. Masson, Paris.
- [Waltz and Miller, 1997] F. M. Waltz and J. W. V. Miller, 1997. An efficient algorithm for Gaussian blur using finite-state machines. *SPIE Conf. on Machine Vision System for Inspection and Metrology*, vol. 3521.
- [Wang and al., 2005] W. Wang, H. Pottmann and Y. Liu, 2005. Fitting B-spline curves to point clouds by squared distance minimization. *ACM Transactions on Graphics*. To appear.
- [Woo and al., 1999] M. Woo, J. Neider, T. Davis and D. Shreiner, 1999. OPENGL® PROGRAMMING GUIDE: THE OFFICIAL GUIDE TO LEARNING OPENGL, VERSION 1.2 (3RD EDITION). Addison-Wesley.

- [Wyvill and van Overveld, 1996] B. Wyvill and K. van Overveld, 1996. Polygonization of Implicit Surfaces with Constructive Solid Geometry. *Journal of Shape Modelling*, vol. 2, n°4, pages 257–274.
- [Yang and al., 2004] H. Yang, W. Wang and J. Sun, 2004. Control point adjustment for B-spline curve approximation. *Computer-Aided Design*, vol. 36, n°7, pages 639–652.
- [Yoshizawa and al., 2002] S. Yoshizawa, A. G. Belyaev and H.-P. Seidel, 2002. A simple approach to interactive free-form shape deformations. *Pacific Graphics*, pages 471–474.
- [Yoshizawa and al., 2003] S. Yoshizawa, A. G. Belyaev and H.-P. Seidel, 2003. Free-form skeleton-driven mesh deformations. *ACM Solid Modeling*, pages 247–253.

DECLARATIVE APPROACH OF SURFACES MODELLING

We are interested in the design of surfaces, essentially NURBS, most often considered to model objects. One advantage of these surfaces is to design a form according to the position of control points. The declarative approach of surfaces modelling is developed as a fast and easy way to realize sketches of forms and surfaces. It is also specially devoted to speed up the specialists' design giving them one or several solutions satisfying constraints and properties.

The work achieved during this PhD. is divided into four parts :

1. Feasibility study : it was carried out with the *École Nationale des Arts et Métiers* in Aix-en-Provence. The study was focused on description and on modelling of machine parts.
2. Conceptual analysis : This stage, which is fundamental on large projects, enabled us to design the global architecture of our declarative process of surfaces. In order to develop a first prototype we decided to focus on the resolution side divided into two parts : research of solution classes and their construction under constraints.
3. Study and development of the "research of solutions classe" part : our approach is based on the behaviour of the surface deformations during its construction phase : two surfaces belong to the same class if their behaviours are similar during the same sequence of deformations.
4. Study and development of the "construction under constraints" part : In order to obtain a specific solution surface (or an instance of solution class) we chose a construction method based on successive deformations. These deformations are achieved under more or less strong constraints. Thus, we developed a surface deformation method able to satisfy matching constraints which keep a well-controlled shape of each influence area.

Two applications implementing this work have been developed using the C++ language and are available under MS Windows[®] 32-bit versions, Linux and MacOS X.

Key words: Geometric modeling, Declarative, Surface, NURBS, semantics, CAD, Geometric constraints.

Academic discipline: Computer Science.

Laboratory: LSIS - Marseille - France (Laboratory of Information Science and Systems).

APPROCHE DÉCLARATIVE DE LA MODÉLISATION DE SURFACES

Nous nous intéressons à la création de surfaces à pôles (NURBS essentiellement), largement utilisées dans les systèmes de modélisation géométrique. Un des avantages de cette modélisation est de permettre d'appréhender la forme des surfaces par la position de points de contrôle. L'approche déclarative de la modélisation de surfaces est destinée à la réalisation rapide et facile d'ébauches de formes et de surfaces. Elle est aussi et surtout destinée à accélérer les processus de conception des spécialistes en leur proposant des solutions adaptées répondant à un ensemble de contraintes et de propriétés.

Pour y parvenir, le travail réalisé dans la thèse se divise en quatre étapes :

1. Étude de faisabilité : réalisée en collaboration avec l'École Nationale Supérieure des Arts et Métiers d'Aix-en-Provence, elle s'est focalisée sur la description et la modélisation d'objets de type pièces mécaniques.
2. Analyse conceptuelle : Cette étape primordiale dans un projet d'une telle envergure nous a permis de mettre au point l'architecture générale de notre processus déclaratif de surfaces. Dans le cadre de la thèse et en vue d'un premier prototype, nous décidons de focaliser notre étude sur la partie résolution qui se scinde en deux : la détermination des classes de solutions et leur construction sous contraintes.
3. Étude et développement de la «détermination des classes de solutions» : Notre approche repose sur le comportement d'une surface face aux déformations qui lui sont apportées lors de sa construction : deux surfaces appartiennent à la même classe si elles ont le même comportement face à la même succession de déformations.
4. Étude et développement de la «construction sous contraintes» : Afin d'obtenir une surface solution particulière (ou instance d'une classe de solutions) nous choisissons une construction par l'application successive de déformations. Ces déformations sont soumises à des contraintes plus ou moins fortes. Nous avons donc élaboré une méthode de déformations de surfaces capable de satisfaire des contraintes de passage tout en gardant un fort contrôle sur la forme de chaque zone d'influence.

Deux applications mettant en œuvre ce travail ont été réalisées en C++ et sont disponibles sous les versions 32-bits de MS Windows®, Linux et MacOS X.

Mots clefs : Modélisation géométrique, Déclaratif, Surface, NURBS, Sémantique, CAO, Contraintes géométriques.

Discipline : Informatique.

Laboratoire : LSIS - Marseille - France (Laboratoire des Sciences de l'Information et des Systèmes).